New Nonlinear Machine Learning Algorithms with Applications to Biomedical Data Science

by

Xiaoqian Wang

BS, Zhejiang University, 2013

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2019

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Xiaoqian Wang

It was defended on

May 31, 2019

and approved by

Heng Huang, PhD, John A. Jurenko Endowed Professor, Department of Electrical and Computer Engineering

Zhi-Hong Mao, PhD, Professor, Department of Electrical and Computer Engineering

Wei Gao, PhD, Associate Professor, Department of Electrical and Computer Engineering

Jingtong Hu, PhD, Assistant Professor, Department of Electrical and Computer Engineering

Jian Ma, PhD, Associate Professor, Computational Biology Department, Carnegie Mellon University

Dissertation Director: Heng Huang, PhD, John A. Jurenko Endowed Professor, Department of Electrical and Computer Engineering

New Nonlinear Machine Learning Algorithms with Applications to Biomedical Data Science

Xiaoqian Wang, PhD

University of Pittsburgh, 2019

Recent advances in machine learning have spawned innovation and prosperity in various fields. In machine learning models, nonlinearity facilitates more flexibility and ability to better fit the data. However, the improved model flexibility is often accompanied by challenges such as overfitting, higher computational complexity, and less interpretability. Thus, it is an important problem of how to design new feasible nonlinear machine learning models to address the above different challenges posed by various data scales, and bringing new discoveries in both theory and applications. In this thesis, we propose several newly designed nonlinear machine learning algorithms, such as additive models and deep learning methods, to address these challenges and validate the new models via the emerging biomedical applications.

First, we introduce new interpretable additive models for regression and classification and address the overfitting problem of nonlinear models in small and medium scale data. we derive the model convergence rate under mild conditions in the hypothesis space and uncover new potential biomarkers in Alzheimer's disease study. Second, we propose a deep generative adversarial network to analyze the temporal correlation structure in longitudinal data and achieve state-of-the-art performance in Alzheimer's early diagnosis. Meanwhile, we design a new interpretable neural network model to improve the interpretability of the results of deep learning methods. Further, to tackle the insufficient labeled data in large-scale data analysis, we design a novel semi-supervised deep learning model and validate the performance in the application of gene expression inference.

Table of Contents

List of Tables

List of Figures

Preface

The past six years of PhD study has been the best journey I could have in my life. I feel full heart of gratitude as I start writing this thesis. Especially for my doctoral supervisor, Dr. Huang Heng, I sincerely thank him, since I will never have the opportunity to dive into the field of machine learning research without his guidance, trust and strong support. Dr. Huang is a great idol in our lab for being highly self-motivated and confident. We are always deeply impressed by his great passion for research and education. It is Dr. Huang that provided me with a picture of the life of top machine learning researcher, and the main reason to pursue an academic career for my future life. I am really lucky that I can become the student of Dr. Huang and get his guidance along my path toward research and career.

I would also like to thank Dr. Zhi-Hong Mao, Dr. Wei Gao, Dr. Jingtong Hu, and Dr. Jian Ma for being on my PhD committee. I really appreciate the great advice and guidance for the research direction and for academic job hunting. I feel honored to get the inspiring instructions and am grateful to the committee members for their time and help.

Thanks to every member in the CSL lab. Many thanks to Dr. Feiping Nie and Dr. Hong Chen for their guidance and collaboration during my PhD study. Thanks to Kamran, Hongchang, Zhouyuan, Guodong, Feihu, Bin, Lei, Yanfu, Runxue, and Haoteng for the discussion about research. I have learned so many things from all members in the CSL lab. I am very lucky that I could be in such a wonderful research group and get the opportunity to make friends and work together with so many great people.

Thanks to my friends, Jia, Akshay, Jie Tang, Xi, Yong, Amy, Jie Xu, Yizhuo, Kun, Zhihui, Yini, Xu, and Mengjiao. Thanks for the company and care. I feel so fortune to be surrounded by good friends who listen to me, understand me, and bring the sunshine in my life. Thanks for making my PhD life colorful and enjoyable and let me feel always supported.

During my job hunting in academia, I have visited over 10 universities and meet with so many great professors. I feel extremely welcomed by the very kind people and am sincerely thankful to Xingbo Wu, Brian Ziebart, Jason Wang, Grace Wang, Xiangnan Kong, Jian Zou, Sudhir Kumar, Zoran Obradovic, Slobodan Vucetic, Haibin Ling, Yuying Xie, Andrew Christlieb, Pang-Ning Tan,

Douglas Schmidt, Stanley Chan, Felix Lin, Saurabh Bagchi, Charles A. Bouman, Milind Kulkarni, Yung-Hsiang Lu, Predrag Radivojac, David Miller, Qun Li, Chao Chen, Steven Skiena, Zhi Ding, and many others. I am grateful to have the opportunity to talk with the professors and learn about how to start the academic career.

Special thanks to my parents, Yong and Qingwei. I could never become who I am without their love and support. Thanks for teaching me to be an upright person and to follow my own enthusiasm and passion. Sincere thanks to my boyfriend, Guohua. Thanks for all the understanding and caring in the past years. He is always my mentor and a great friend. I could never have the courage to take the job at Purdue University without his support. Thanks for always being very patient, caring, and supportive for me during the PhD journey.

1.0 Introduction

1.1 Background

In recent decades, machine learning has achieved unparalleled success in various fields, from automatic translation, object detection to autonomous driving and computer-aided diagnosis. Compared with linear methods that hold the assumption of linear association between data, nonlinear models, such as kernels methods, and deep neural network has exhibited a high degree of flexibility and representative power thus has better generalization power in data with complex structures such as images, natural languages, videos and biomedical data.

Despite the wide application and great success, challenges still remain in how to design a good machine learning model. For small to medium scale data, deep learning methods suffer from overfitting problem thus face the problem of bad generalization performance due to training of a model with high variance with limited number of training samples. As for kernel methods, the training involves a large kernel matrix thus introduces high order of computational complexity.

For the training of nonlinear machine learning methods in large-scale data, deep learning models require a large amount of labeled data to build an inference network with satisfying performance. However, the collection of large libraries of labeled data is still difficult and expensive. In many machine learning applications, it typically requires great effort of human annotators or expensive special devices to label samples. On the contrary, unlabeled data is much easier and less expensive to achieve. To guarantee the quality of the model while reduce the cost of acquiring labels, lots of effort has been made on semi-supervised learning such that the unlabeled data can be better utilized to strengthen the performance of supervised learning. Based on the difference in how to incorporate the information from unlabeled data, semi-supervised classification models can be divided into different types, such as self-training, generative model, co-training, semi-supervised SVM, graph-based methods, *etc*. See [177, 86] for detailed background on semi-supervised learning models.

Moreover, deep learning models with state-of-the-art performance are usually formulated as black boxes, making it difficult to interpret how the models make decisions. In areas such as health care, autonomous driving and job hiring, human end-users cannot blindly and faithfully trust the predictions from a well-performed model without knowing the mechanism behind the model behavior. The poor interpretability of black-box models can trigger trust issues from human users. It remains an important problem of how to properly address the above challenges when designing new machine learning models for different scale of data in order to improve the trustworthiness and performance of the model.

On the other hand, the rapid development of high-throughput technology has spawned detailed molecular, cellular and pathological features of many diseases, which enables deep and systematic study in medical research. The availability of comprehensive biological data not only helps researchers around the world to share genomic and clinical information, but also helps to analyze molecular characteristics of serious diseases that threaten human health.

In addition to unprecedented opportunities to uncover vast amounts of biological knowledge, the emergence of large-scale biological data has also brought enormous challenges to processing and interpretation of data. With a large amount of biological data, there is an urgent need for reliable computing models to integrate complex information in an accurate and efficient manner.

For example, Alzheimer's Disease (AD) is the most common form of dementia, which triggers memory, thinking, and behavior problems. The genetic causal relationship of AD is complex [7] and therefore presents difficulties in the prevention, diagnosis and treatment of this disease. Recent advances in multimodal neuroimaging and high throughput genotyping and sequencing techniques bring an emerging research field, imaging genomics, which provides exciting new opportunities to ultimately improve our understanding of brain disease, their genetic architecture, and their influences on cognition and behavior.

The rapid progress in neuroimaging techniques has provided insights into early detection and tracking of neurological disorders [157]. Later research interest in imaging neuroscience has focused on Genome Wide Association Studies (GWAS) to examine the association between genetic markers, called Single Nucleotide Polymorphisms (SNPs), and imaging phenotypes [154, 27], with the goal of finding explanations for the variability observed in brain structures and functions. However, these research works typically study associations between individual SNPs and individ-

ual phenotypes and overlook interrelated structures among them. To better understand the genetic causal factors of brain imaging abnormalities, previous works have laid great emphasis on identifying relevant QTL [144, 118], which related high-throughput SNPs to imaging data and enhanced the progress and prosperity of neuroscience research. Besides, extensive work has been proposed to predict MCI conversion using neuroimaging data [123, 55]. Previous methods usually formulate MCI conversion prediction as a binary classification (distinguishing MCI converters from non-converters) [123, 156] or multi-class classification problem (when considering other classes such as AD or health control (HC)) [55, 150], where the methods take the neuroimaging data at baseline time as the input and classify if the MCI samples will convert to AD in years.

The development of efficient and effective computational models can greatly improve understanding of complex human diseases. For example, a regression model can be used to reveal the association between different disease data type. Clustering models reveals the relationship between different patients, thereby promoting the development of individualized personalized medicine.

To facilitate the prosperity in Alzheimer's study, ADNI collects abundant neuroimaing, genetic and cognitive tests data for the study of this disease. Another focus of this thesis is to propose appropriate models to learn the association between different types of data such that the genetic basis as well as biological mechanisms of brain structure and function can be extracted. In addition, since AD is a progressive neurodegenerative disorder, we propose to look into the information conveyed in the longitudinal data. Specifically, it is inspiring to study the prodromal stage of Alzheimer's, MCI, which exhibits a good chance of converting to AD. In all the above studies, we propose to automatically find predominant genetic features/regions of interests (ROIs) which are responsible for the development of AD. Such studies can provide reference for disease diagnosis and drug design.

The goal of this thesis is to put forward novel computational models to address the challenges in designing new machine learning models for different scales of data, and utilize these new models to enhance the understanding in the genesis and progression of severe diseases like Alzheimer's disease.

1.2 Contribution

We summarize our contribution as follows:

- We propose a novel efficient additive model (FNAM) to address the overfitting problem of nonlinear machine learning in small to medium scale data. We provide rigorous theoretical generalization error analysis on our model. In particular, different from conventional analysis with independent samples, our error bound is under m-dependent observations, which is a more general assumption and more appropriate for the high-throughput complex genotypes and phenotypes.
- A new group sparse nonlinear classification algorithm (GroupSAM) is proposed by extending the previous additive regression models to the classification setting, which contains the LPSVM with additive kernel as its special setting. To the best of our knowledge, this is the first algorithmic exploration of additive classification models with group sparsity.
- Theoretical analysis and empirical evaluations on generalization ability are presented to support the effectiveness of GroupSAM. Based on constructive analysis on the hypothesis error, we get the estimate on the excess generalization error, which shows that our GroupSAM model can achieve the fast convergence rate $O(n^{-1})$ under mild conditions. Experimental results demonstrate the competitive performance of GroupSAM over the related methods on both simulated and real data.
- We propose a novel semi-supervised regression framework based on generative adversarial network (GAN) to address the lack of labeled data in large-scale machine learning problem. A new strategy is introduced to stabilize the training of GAN model with high-dimensional output.
- We propose the first longitudinal deep learning model for MCI conversion prediction and achieve state-of-the-art performance in Alzheimer's early detection.

1.3 Notation

Throughout this thesis, unless specified otherwise, upper case letters denote matrices, *e.g.,* $X,\ Y$. Bold lower case letters denote vectors, *e.g.,* $\mathbf{w},\ \mathbf{b}$. Plain lower case letters denote scalars, *e.g.,* $a,\ \gamma$. w_i denotes the i-th element of vector $\mathbf{w}$. $\mathbf{w}_i$ denotes the i-th row of matrix W. $\mathbf{w}^j$ or $\mathbf{w}^{(j)}$denotes the j-th column of W. w_{ij} denotes the ij-th element of matrix W. $\|\mathbf{w}\|_2$ or $\|\mathbf{w}\|$ denotes the ℓ_2-norm of vector $\mathbf{w}$: $\sqrt{\sum_i w_i^2}$. $\|W\|_F$ denotes the Frobenius norm of matrix W: $\|W\|_F = \sqrt{\sum_i \sum_j w_{ij}^2} = \sqrt{\sum_i \|\mathbf{w}_i\|^2}$. $\|W\|_1$ denotes the ℓ_1 norm: $\|W\|_1 = \sum_i \sum_j |w_{ij}|$. $\|W\|_*$ denotes the trace norm (*a.k.a.* nuclear norm): $\|W\|_* = \sum_i \sigma_i$, where σ_i is the i-th singular value of W.

Specially, d denotes the dimension of the feature vector, *i.e.,* number of SNPs. n denotes the number of patients. c represents the number of QTs. $X = [\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_n]^\mathsf{T} \in \mathbb{R}^{n\times d}$ denotes the input SNP matrix, where each row of X represents the genetic variants of each patient. $Y = [\mathbf{y}_1,\ \mathbf{y}_2,\ \ldots,\ \mathbf{y}_n]^\mathsf{T} \in \mathbb{R}^{n\times c}$ represents the input imaging feature matrix where each row of Y denotes the phenotype of one patient. I stands for the identity matrix, and $\mathbf{1}$ stands for a vector with all elements being 1.

1.4 Proposal Organization

The rest of the proposal is organized as follows. In Chapter 2, we introduce a new additive model for addressing the challenges of overfitting and less interpretability in nonlinear machine for small to medium scale data. We propose the model in an efficient manner with the time complexity the same as linear models and also provide theoretical analysis on the convergence properties of the model. In Chapter 3, we answer the question of how to analyze the feature interaction in additive model with a new additive classification model. We proved the convergence property of our new model, which has a satisfactory learning rate with polynomial decay. We conduct extensive experiments on synthetic and real data to validate the model performance. In Chapter 4 we propose two new deep learning models for addressing the problem of lacking labeled data in non-linear ma-

chine learning. We propose the two new models on the basis of generative adversarial network and improve the model performance by effectively learning from the labeled data and large amount of unlabeled data. We show extensive results in the application of gene expression inference to validate the performance. In Chapter 5 we design a new generative model to analyze the longitudinal structure in the data and achieve state-of-the-art performance in MCI conversion prediction. Next in Chapter 6 we propose a new idea to build self-explaining deep neural network via additive model and validate the performance in MCI conversion prediction. Finally, we conclude the thesis in Chapter 7 and propose some open problems and future direction.

2.0 Additive Model for Small/Medium Scale Data

2.1 Motivation

In previous works, several machine learning models were established to depict the relations between SNPs and brain endophenotypes [148, 162, 176, 151, 59]. In [148, 176, 151, 59], the authors used the low-rank learning models or structured sparse learning models to select the imaging features that share common effects in the regression analysis. [162] applied the LASSO regression model to discover the significant SNPs that are associated with brain imaging features. However, previous works use linear models to predict the relations between genetic biomarkers and brain endophenotypes, which may introduce high bias during the learning process. Since the influence of QTL is complex, it is crucial to design appropriate non-linear model to investigate the genetic biomarkers (due to the limited size of biological data, deep learning models don't work well for our problem). Besides, most previous computational models on genotype and phenotype studies did not provide theoretical analysis on the performance of the models, thus leaves uncertainty in the validity of the models.

To tackle with these challenging problems, in this chapter, we propose a novel and efficient nonlinear model for the identification of QTL. We apply our model to the QTL identification of Alzheimer's disease (AD), the most common cause of dementia. By means of feedforward neural networks, our model can be flexibly employed to explain the non-linear associations between genetic biomarkers and brain endophenotypes, which is more adaptive for the complicated distribution of the high-throughput biological data. We would like to emphasize the following contributions of our work:

- We propose a novel additive model with generalization error analysis. In particular, different from conventional analysis with independent samples, our error bound is under m-dependent observations, which is a more general assumption and more appropriate for the high-throughput complex genotypes and phenotypes.

- Our model is efficient in computation. The time complexity of our model is linear to the number of samples and number of features in the data. Experimentally we showed that it only takes a few minutes to run our model on the ADNI data.
- Experimental results demonstrate that our model not only identifies several well-established AD-associated genetic variants, but also finds out new potential SNPs.

2.2 Related Work

In QTL identification of brain imaging abnormalities, the goal is to learn a prediction function which estimates the imaging feature matrix $Y = [\mathbf{y}_1,\ \mathbf{y}_2,\ \ldots,\ \mathbf{y}_n]^\mathsf{T} \in \mathbb{R}^{n\times c}$ given the genetic information $X = [\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots,\ \mathbf{x}_n]^\mathsf{T} \in \mathbb{R}^{n\times d}$. Meanwhile, we want to weigh the importance of each SNP in the prediction according to the learning model. The most straightforward method is least square regression, which learns a weight matrix $W \in \mathbb{R}^{d\times c}$ to study the relations between SNPs and brain endophenotypes. W is an intuitive reflect of the importance of each SNP for the prediction of each endophenotype.

Based on least square regression, several models were proposed for QTL identification. In [139, 162], the authors employed sparse regression models for the discovery of predominant genetic features. In [40, 148], low-rank constraint was imposed to uncover the group structure among SNPs in the association study.

In the identification of QTL, previous works mainly use linear models for the prediction. However, according to previous studies, the biological impact of genetic variations is complex [95] and the genetic influence on brain structure is complicated [108]. Thus, the relations between genetic biomarkers and brain-imaging features may not be necessarily linear and the prediction with linear models is likely to trigger large bias.

To depict the non-linear association between genetic variations and endophenotypes, neural networks introduce a convenient and popular framework. [122] proposed feed forward neural networks with random weights (FNNRW), which can be formed as:

$$f(\mathbf{x}) = \sum_{t=1}^{h} a_t \phi(\langle \mathbf{v}_t, \mathbf{x}\rangle + b_t)\,, \tag{2.1}$$

where $\mathbf{x} = [x_1,\ x_2,\ \ldots,\ x_d] \in \mathbb{R}^d$ is the input data, h is the number of hidden nodes, $\mathbf{v}_t|_{t=1}^h = [v_{t1},\ v_{t2},\ \ldots,\ v_{td}] \in \mathbb{R}^d$ is the parameter in the hidden layer for t-th hidden node, $b_t \in \mathbb{R}$ is the corresponding bias term, $\langle \mathbf{v}_t, \mathbf{x} \rangle = \sum_{j=1}^{d} v_{tj} x_j$ represents Euclidean inner product, $\phi(.)$ is the activation function, and $a_t \in \mathbb{R}$ is the weight for the t-th hidden node.

As is analyzed in [60, 112], FNNRW enjoys an obvious advantage in computational efficiency over neural nets with back propagation. In Eq. (2.1), $\mathbf{v}_t$ and b_t are randomly and independently chosen before hand, and the randomization in parameter largely relieves the computational burden. FNNRW is aimed at estimating only the weight parameter $a_t|_{t=1}^h$ thus is extremely efficient. Such property makes FNNRW more appropriate for analysis of the high-throughput data in Alzheimer's research.

[112] constructed a classifier using FNNRW where they conduct classification on the featurized data as shown in Eq. (2.1). The classification model can be easily extended to the regression scenario with the objective function formulated as:

$$\min_{\mathbf{a}_t|_{t=1}^h} \left\| Y - \sum_{t=1}^{h} \phi(X\mathbf{v}_t^{\mathsf{T}} + b_t\mathbf{1})\mathbf{a}_t \right\|_F^2 + \gamma \sum_{t=1}^{h} \|\mathbf{a}_t\|_2^2 \ , \tag{2.2}$$

where γ is the hyper-parameter for the regularization term and $\mathbf{a}_t = [a_1,\ a_2,\ \ldots,\ a_c] \in \mathbb{R}^c$ is the weight parameter of the t-th hidden node for c different endophenotypes. As discussed above, Problem (2.2) can be adopted to efficiently estimate the nonlinear associations between genetic variations and brain endophenotypes. However, since the parameters of hidden layer is randomly assigned, traditional FNNRW model makes it hard to evaluate the importance of each feature.

To tackle with these problems, we propose a novel additive model in next section, which not only maintains the advantage of computational efficiency of FNNRW but also integrates the flexibility and interpretability of additive models.

2.3 FNAM for Quantitative Trait Loci Identification

We propose new Additive Model via Feedforward Neural networks with random weights (FNAM) as:

$$f_a(X) = \sum_{t=1}^{h} \sum_{j=1}^{d} \phi(v_{tj}\mathbf{x}_j + b_t\mathbf{1})\mathbf{a}_t \,, \tag{2.3}$$

where we distinguish the contribution of each feature $\mathbf{x}_j$ and formulate the model in an additive style for the prediction. Similar to that of FNNRW, we propose to optimize the least square loss between the ground truth endophenotype matrix Y and the estimation $f_a(X)$ with ℓ_2-norm penalization, then we propose the following objective function:

$$\min_{\mathbf{a}_t|_{t=1}^{h}} \left\| Y - \sum_{t=1}^{h} \sum_{j=1}^{d} \phi(v_{tj}\mathbf{x}_j + b_t\mathbf{1})\mathbf{a}_t \right\|_F^2 + \gamma \sum_{t=1}^{h} \|\mathbf{a}_t\|_2^2 \,, \tag{2.4}$$

For simplicity, if we define $A = [\mathbf{a}_1,\ \mathbf{a}_2,\ \ldots,\ \mathbf{a}_h]^\mathsf{T} \in \mathbb{R}^{h\times c}$ as the weight parameter for hidden nodes, and $G \in \mathbb{R}^{n\times h}$ such that

$$G = \begin{bmatrix} \sum_{j=1}^{d} \phi(v_{1j}x_{1j} + b_1) & \ldots & \sum_{j=1}^{d} \phi(v_{hj}x_{1j} + b_h) \\ \vdots & \ldots & \vdots \\ \sum_{j=1}^{d} \phi(v_{1j}x_{nj} + b_1) & \ldots & \sum_{j=1}^{d} \phi(v_{hj}x_{nj} + b_h) \end{bmatrix} , \tag{2.5}$$

then we could rewrite our objective function Problem (2.4) as:

$$\min_{A} \|Y - GA\|_F^2 + \gamma \|A\|_F^2 \,. \tag{2.6}$$

Take derivative *w.r.t.* A in Problem (2.6) and set it to 0, we get the closed form solution of A as below:

$$A = (G^\mathsf{T}G + \gamma I)^{-1}G^\mathsf{T}Y \,. \tag{2.7}$$

As discussed in the previous section, one obvious advantage of FNAM over FNNRW is that FNAM considers the role of each feature independently in the prediction, thus makes it possible to interpret the importance of each SNP in the identification QTL, which is a fundamental goal of jointly studying genetic and brain imaging features.

Algorithm 1 Optimization Algorithm of FNAM for QTL Identification.

Input:

SNP matrix $X \in \mathbb{R}^{n\times d}$, endophenotype $Y \in \mathbb{R}^{n\times c}$, number of hidden nodes h, parameter γ.

Output:

Weight matrix $A \in \mathbb{R}^{h\times c}$ for the hidden nodes. Weight matrix $W \in \mathbb{R}^{d\times c}$ showing the relative importance of the d SNPs in the prediction.

1: **Initialize** the weight matrix $V \in \mathbb{R}^{h\times d}$ randomly according to uniform distribution $\mathcal{U}(0,\ 1)$.

2: **Initialize** the bias term $b \in \mathbb{R}^{h}$ randomly according to uniform distribution $\mathcal{U}(0,\ 1)$.

3: 1. Compute G matrix according to the definition in Eq. (2.5).

4: 2. Update A according to the solution in Eq. (2.7)

5: 3. Compute W according to the definition in Eq. (2.9).

Here we discuss how to estimate the role of each feature in FNAM. To separate the contribution of each feature, we rewrite Eq. (2.3) as below:

$$f_a(X) = \sum_{j=1}^{d}(\sum_{t=1}^{h}\phi(v_{tj}\mathbf{x}_j + b_t\mathbf{1})\mathbf{a}_t)\,, \tag{2.8}$$

which indicates that the prediction function $f_a(X)$ can be regarded as the summation of d terms, where the j-th term $\sum_{t=1}^{h}\phi(v_{tj}\mathbf{x}_j + b_t\mathbf{1})\mathbf{a}_t$ denotes the contribution of the j-th feature.

Naturally, if we normalize the magnitude of the j-th term with the ℓ_2-norm of $\mathbf{x}_j$, we could get a good estimation of the significance of the j-th feature. As a consequence, we could define a weight matrix $W \in \mathbb{R}^{d\times c}$ to show the importance of the d SNPs in the prediction of the c imaging features respectively, such that:

$$w_{jl} = \frac{\left\|\sum_{t=1}^{h}\phi(v_{tj}\mathbf{x}_j + b_t\mathbf{1})a_{tl}\right\|}{\|\mathbf{x}_j\|}, j = 1,\dots d,\ \ l = 1,\dots c\,, \tag{2.9}$$

We summarize the optimization step of AFNNRW in Algorithm 1 and provide rigorous convergence proof of AFNNRW in the Appendix.

Time Complexity Analysis: We summarize the optimization steps of FNAM in Algorithm 1. In Algorithm 1, the time complexity of Step 1 (computing G) is $O(ndh)$, the time complexity of Step 2 (computing A) is $O(h^2n + hnc)$, and the time complexity of Step 3 (computing W) is

$O(ndhc)$, where n is the number of patients, d denotes the number of SNPs, and c represents the number of brain endophenotypes. Typically, we have $d > h$ and $d > c$ in the identification of QTL, thus the total time complexity of Algorithm 1 is $O(ndhc)$.

2.4 Generalization Ability Analysis

In this section, based on the real situation of biological data, we provide theoretical analysis on the approximation ability of our FNAM model and derive the upper bound of generalization error.

In most previous works, theoretical analysis is based on the hypothesis of independent and identically distributed (i.i.d.) samples. However, the i.i.d. sampling is a very restrictive concept that occurs only in the ideal case. As we know, the acquisition of high-throughput biological data involves complicated equipments, reagents as well as precise operation of highly trained technicians, which usually introduce variations to the data during the measurement process [77]. Thus, the i.i.d. sampling assumption is not appropriate for the high-throughput biological data analysis. In this section, we provide a learning rate estimate of our model in a much general setting, *i.e.,* m-dependent observations [96].

For simplicity, here we consider the prediction of only one brain endophenotype $\mathbf{y} \in \mathbb{R}^n$, which could be easily extended to the case with multiple endophenotypes. Besides, we incorporate the bias term $\mathbf{b}$ into the weight matrix V by adding one feature valued 1 for all samples to the data matrix X. For analysis feasibility, we reformulate the general FNAM model as below.

Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is a compact metric space and $\mathcal{Y} \subset [-k,\ k]$ for some constant $k > 0$. For any given $\mathbf{z} = \{(\mathbf{x}_i,\ y_i)\}_{i=1}^n \in \mathcal{Z}^n$ and each $j \in \{1,\ 2, \ldots,\ d\}$, we denote $\phi_i^{(j)} = [\phi(v_{1j}, x_{ij}),\ \ldots,\ \phi(v_{hj}, x_{ij})]^\mathsf{T} \in \mathbb{R}^h$ and $\mathbf{v}^{(j)} = [v_{1j},\ v_{2j},\ \ldots,\ v_{hj}]^\mathsf{T} \in \mathbb{R}^h$, where each v_{tj}, $1 \le t \le h$, is generated i.i.d. from a distribution μ on $[0,\ 1]$.

The FNN with random weights in FNAM can be formulated as the following optimization problem:

$$\mathbf{a}_\mathbf{z} = \underset{\mathbf{a}\in\mathbb{R}^{hd}}{\arg\min} \left\{ \frac{1}{n} \sum_{i=1}^{n} \Big(\sum_{j=1}^{d} (\mathbf{a}^{(j)})^\mathsf{T} \phi_i^{(j)} - y_i \Big)^2 + \gamma \sum_{j=1}^{d} \left\| \mathbf{a}^{(j)} \right\|_2^2 \right\}, \tag{2.10}$$

where $\mathbf{a}^{(j)} = [a_1^{(j)}, a_2^{(j)}, ..., a_h^{(j)}]^\mathsf{T} \in \mathbb{R}^h$.

The predictor of FNAM is $f_{\mathbf{z}} = \sum_{j=1}^{d}\sum_{t=1}^{h} a_{\mathbf{z},t}^{(j)}\phi(v_{tj},\cdot)$, to investigate the generalization error bound of FNAM, we rewrite it from a function approximation viewpoint.

Define the hypothesis function space of FNAM as:

$$\mathcal{M}_h = \left\{ f = \sum_{j=1}^{d} f^{(j)} : f^{(j)} = \sum_{t=1}^{h} a_{tj}\phi(v_{tj},\cdot), a_{tj} \in \mathbb{R} \right\} \tag{2.11}$$

and for any $j \in \{1,\ 2,\ \ldots,\ d\}$

$$\left\| f^{(j)} \right\|_{\ell_2}^2 = \inf\left\{ \left\| \mathbf{a}^{(j)} \right\|_2^2 : f = \sum_{t=1}^{h} a_{tj}\phi(v_{tj},\cdot) \right\}. \tag{2.12}$$

Then, FNAM can be rewritten as the following optimization problem:

$$f_{\mathbf{z}} = \sum_{j=1}^{d} f_{\mathbf{z}}^{(j)} = \arg\min_{f\in\mathcal{M}_h} \left\{ \mathcal{E}_{\mathbf{z}}(f) + \gamma \sum_{j=1}^{d} \left\| f^{(j)} \right\|_{\ell_2}^2 \right\}, \tag{2.13}$$

where $\mathcal{E}_{\mathbf{z}}(f)$ is the empirical risk defined by $\mathcal{E}_{\mathbf{z}}(f) = \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i) - y_i)^2$.

For the regression problem, the goal of learning is to find a prediction function $f : \mathbf{x} \to \mathbb{R}$ such that the expected risk

$$\mathcal{E}(f) = \int_{\mathcal{Z}} (y - f(\mathbf{x}))^2 d\rho(\mathbf{x}, y) \tag{2.14}$$

is as small as possible. It is well known that the Bayes function

$$f_\rho(\mathbf{x}) = \int_{\mathcal{Y}} y d\rho(y|\mathbf{x}) \tag{2.15}$$

is the minimizer of $\mathcal{E}(f)$ over all measurable functions. Therefore, the excess expected risk $\mathcal{E}(f) - \mathcal{E}(f_\rho)$ is used as the measure to evaluate the performance of learning algorithm.

Since $\mathcal{Y} \subset [-k, k]$ and $\|f_\rho\|_\infty \leq k$, we introduce the clipping operation

$$\pi(f) = \max(-k, \min(f(\mathbf{x}), k)) \tag{2.16}$$

to get tight estimate on the excess risk of FNAM. Recall that FNAM in (2.4) depends on the additive structure and random weighted networks. Indeed, theoretical analysis of standard random weighted networks has been provided in [60, 112] to characterize its generalization error bound. However, the previous works are restricted to the setting of i.i.d. samples, and do not cover the additive models. Hence, it is necessary to establish the upper bound of $\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho)$ with

much general setting, *e.g.*, m-dependent observations [96, 143]. In this work we consider this more general condition, m-dependent observations other than $i.i.d.$ condition such that the model is applicable to more application problems.

Now, we introduce some necessary definitions and notations for theoretical analysis. Let $\{Z_i = (X_i, Y_i)\}_{i=1}^{\infty}$ be a stationary random process on a probability space $(\Omega, \mathcal{A}, P)$. Denote $\mathcal{A}_1^i$ as the σ-algebras of events generated by $(Z_1,\ Z_2,\ \ldots,\ Z_i)$ and denote $\mathcal{A}_{i+m}^{\infty}$ as the σ-algebras of events generated by $(Z_{i+m},\ Z_{i+m+1},\ \ldots)$.

Definition. For $m \geq 0$, if $\mathcal{A}_1^i$ and $\mathcal{A}_{i+m}^{\infty}$ are independent, we call $\{\mathcal{Z}_i\}_{i=1}^{\infty}$ m-dependent.

It is clear that $m = 0$ for i.i.d. observations.

It is a position to present the main result on the excess risk $\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho)$.

Theorem 1. *Let $f_{\mathbf{z}}$ be defined in (2.4) associated with m-dependent observations $\mathbf{z} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$. There holds*

$$\begin{aligned} & E_{\rho^n} E_{\mu^h} \left\| \pi(f_{\mathbf{z}}) - f_\rho \right\|_{L^2_{\rho_{\mathcal{X}}}}^2 \\ \leq\ & c\sqrt{\frac{\log n^{(m)} - \frac{1}{2}\log\gamma}{n^{(m)}}} \\ & + \inf_{f \in \mathcal{M}_h} \Big\{ \Big\| f - f_\rho \Big\|_{L_{\rho_{\mathcal{X}}}}^2 + \gamma \sum_{j=1}^{d} \Big\| f^{(j)} \Big\|_{\ell_2}^2 \Big\}, \end{aligned} \tag{2.17}$$

where $n^{(m)} = \lfloor \frac{n}{m+1} \rfloor$, $\| \cdot \|_{L^2_{\rho_{\mathcal{X}}}}$ is norm of square integral function space $L^2_{\rho_{\mathcal{X}}}$, and c is a positive constant independent of $n^{(m)}, \gamma$.

Theorem 1 demonstrates that FNAM can achieve the learning rate $O(\sqrt{\frac{\log n^{(m)}}{n^{(m)}}})$ as the hypothesis space satisfies

$$\inf_{f \in \mathcal{M}_h} \Big\{ \Big\| f - f_\rho \Big\|_{L^2_{\rho_{\mathcal{X}}}}^2 + \gamma \sum_{j=1}^{d} \Big\| f^{(j)} \Big\|_{\ell_2}^2 \Big\} = O\Big(\sqrt{\frac{\log n^{(m)}}{n^{(m)}}} \Big). \tag{2.18}$$

When $f_\rho \in \mathcal{M}_h$, we have

$$\lim_{n \to \infty} E_{\rho^n} E_{\mu^h} \left\| \pi(f_{\mathbf{z}}) - f_\rho \right\|_{L^2_{\rho_{\mathcal{X}}}}^2 = 0, \tag{2.19}$$

which means the proposed algorithm is consistency. The current result extends the previous theoretical analysis with i.i.d samples [60, 112] to the m-dependent observations. Indeed, we can also obtain the error bound for strong mixing samples by the current analysis framework.

The following Bernstein inequality for m-dependent observations (Theorem 4.2 in [96]) is used for our theoretical analysis.

Lemma 1. *Let $\{Z_i\}_{i=1}^{\infty}$ be a stationary m-dependent process on probability space $(\Omega, \mathcal{A}, P)$. Let $\psi : \mathbb{R} \to \mathbb{R}$ be some measurable function and $U_i = \psi(Z_i), 1 \le i \le \infty$. Assume that $|U_1| \le d_1$ and $EU_1 = 0$. Then, for all $n \ge m+1$ and $\epsilon > 0$,*

$$P\Big\{\frac{1}{n}\sum_{i=1}^{n} U_i \ge \epsilon\Big\} \le \exp\Big\{-\frac{n^{(m)}\epsilon^2}{2(E|U_1|^2 + \frac{\epsilon d_1}{3})}\Big\}, \tag{2.20}$$

where $n^{(m)} = \lfloor \frac{n}{m+1} \rfloor$ is the number of "effective observations".

The covering number is introduced to measure the capacity of hypothesis space, which has been studied extensively in [28, 29, 178].

Definition. The covering number $\mathcal{N}(\mathcal{F}, \epsilon)$ of a function set $\mathcal{F}$ is the minimal integer l such that there exists l disks with radius ϵ covering $\mathcal{F}$.

Considering the hypothesis space $\mathcal{M}_h$ in Section 4, we define its subset

$$\mathcal{B}_R = \Big\{f \in \mathcal{M}_h : \sum_{j=1}^{d} \big\|f^{(j)}\big\|_{\ell_2}^2 := \sum_{j=1}^{d}\sum_{t=1}^{h} |a_{tj}|^2 \le R^2\Big\}. \tag{2.21}$$

Now we present the uniform concentration estimate for $f \in \mathcal{B}_R$

Lemma 2. *Let $\mathbf{z} = \{z_i\}_{i=1}^n := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \in \mathcal{Z}^n$ be m-dependent observations. Then*

$$\begin{aligned} &P\Big\{\sup_{f\in\mathcal{B}_R}\Big(\mathcal{E}(\pi(f)) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f)) - \mathcal{E}_{\mathbf{z}}(f_\rho))\Big) \ge \epsilon\Big\} \\ \le\, &\mathcal{N}(\mathcal{B}_R, \frac{\epsilon}{16k}) \cdot \exp\Big\{-\frac{n^{(m)}\epsilon^2}{512k^2 + 22k\epsilon}\Big\}. \end{aligned} \tag{2.22}$$

Proof: Set $U_i = \psi_f(z_i) = \mathcal{E}(\pi(f)) - \mathcal{E}(f_\rho) - ((y_i - \pi(f)(\mathbf{x}_i))^2 - (y_i - f_\rho(\mathbf{x}_i))^2)$. It is easy to verify that $|U_i| \le 8k^2$ and $EU_i = 0$. From Lemma 1 we obtain, for any given m-dependent samples $z = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \in \mathcal{Z}^n$ and measurable function f,

$$\begin{aligned} &P\Big\{\frac{1}{n}\sum_{i=1}^{n}\psi_f(z_i) \ge \epsilon\Big\} \\ =\;& P\Big\{\mathcal{E}(\pi(f)) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f)) - \mathcal{E}_{\mathbf{z}}(f_\rho)) \ge \epsilon\Big\} \\ \le\;& \exp\Big\{-\frac{n^{(m)}\epsilon^2}{128k^2 + 16k\epsilon/3}\Big\}. \end{aligned} \tag{2.23}$$

Let $J = \mathcal{N}(\mathcal{B}_R, \frac{\epsilon}{16k})$ and $\{f_j\}_{j=1}^J$ be the centers of disks D_j such that $\mathcal{B}_R \subset \bigcup_{j=1}^J D_j$. Observe that, for all $f \in D_j$ and $z \in \mathcal{Z}^n$,

$$\begin{aligned} & \frac{1}{n}\sum_{i=1}^n(\psi_f(z_i) - \psi_{f_j}(z_i)) \\ = & \Big|\mathcal{E}(\pi(f)) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f)) - \mathcal{E}_{\mathbf{z}}(f_\rho)) \\ & -[\mathcal{E}(\pi(f_j)) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f_j)) - \mathcal{E}_{\mathbf{z}}(f_\rho))]\Big| \\ = & \Big|\mathcal{E}(\pi(f)) - \mathcal{E}(f_j) - (\mathcal{E}_{\mathbf{z}}(\pi(f)) - \mathcal{E}_{\mathbf{z}}(f_j))\Big| \\ \leq & \ 8k\left\|f - f_j\right\|_\infty \leq \frac{\epsilon}{2}. \end{aligned} \tag{2.24}$$

It means that

$$\begin{aligned} & \sup_{f\in D_j} \mathcal{E}(\pi(f)) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f)) - \mathcal{E}_{\mathbf{z}}(f_\rho)) \geq \epsilon \\ & \Longrightarrow \mathcal{E}(\pi(f_j)) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f_j)) - \mathcal{E}_{\mathbf{z}}(f_\rho)) \geq \frac{\epsilon}{2}. \end{aligned} \tag{2.25}$$

Then

$$\begin{aligned} & P\Big\{\sup_{f\in\mathcal{B}_R}(\mathcal{E}(\pi(f)) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f)) - \mathcal{E}_{\mathbf{z}}(f_\rho))) \geq \epsilon\Big\} \\ \leq & \sum_{j=1}^J P\Big\{\sup_{f\in D_j}(\mathcal{E}(\pi(f_j)) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f_j)) - \mathcal{E}_{\mathbf{z}}(f_\rho)))\Big\} \\ \leq & \ \mathcal{N}(\mathcal{B}_R, \frac{\epsilon}{16k})\exp\Big\{-\frac{n^{(m)}\epsilon^2}{4(128k^2 + 16k\epsilon/3)}\Big\}. \end{aligned} \tag{2.26}$$

This completes the proof. □

Proof of Theorem 1: According to the definition of $\mathcal{E}(f)$ and f_ρ, we deduce that

$$\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) = \left\|\pi(f_{\mathbf{z}}) - f_\rho\right\|_{L_{\rho\mathcal{X}}}^2 = E_1 + E_2\,, \tag{2.27}$$

where $E_1 = \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) - \mathcal{E}_{\mathbf{z}}(f_\rho))$ and $E_2 = \mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) - \mathcal{E}_{\mathbf{z}}(f_\rho) + \gamma\sum_{j=1}^d \left\|f_{\mathbf{z}}^{(j)}\right\|_{\ell_2}^2$.

Now we turn to bound E_1 in terms of Lemma 2. According to the definition of $f_{\mathbf{z}}$, we get

$$\gamma\sum_{j=1}^d \left\|f_{\mathbf{z}}^{(j)}\right\|_{\ell_2}^2 \leq \mathcal{E}_{\mathbf{z}}(0) \leq k^2\,. \tag{2.28}$$

It means that $f_{\mathbf{z}} \in \mathcal{B}_R$ with $R = \frac{k}{\sqrt{\gamma}}$. By proposition 5 in ([28]), we know that:

$$\log \mathcal{N}(\mathcal{B}_R, \epsilon) \leq hd \log(\frac{4R}{\epsilon}) \,. \tag{2.29}$$

Integrating these facts into Lemma 2, we obtain:

$$\begin{aligned} & P\{E_1 \geq \epsilon\} \\ \leq \; & P\{\sup_{f \in \mathcal{B}_R} (\mathcal{E}(\pi(f)) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f)) - \mathcal{E}_{\mathbf{z}}(f_\rho))) \geq \epsilon\} \\ \leq \; & \exp\Big\{hd \log(\frac{64kR}{\epsilon}) - \frac{n^{(m)}\epsilon^2}{512k^2 + 22k\epsilon}\Big\} \,. \end{aligned} \tag{2.30}$$

Then, for any $\eta \geq \frac{64k^2}{n^{(m)}}$,

$$\begin{aligned} E_{\rho^n}(E_1) &= \int_0^\infty P\{E_1 \geq \epsilon\} d\epsilon \\ &\leq \eta + \int_\eta^\infty \exp\Big\{hd \log(\frac{64k^2}{\epsilon\sqrt{\gamma}}) - \frac{n^{(m)}\epsilon^2}{512k^2 + 22k\epsilon}\Big\} d\epsilon \\ &\leq \eta + \gamma^{-\frac{hd}{2}} \exp\Big\{\frac{n^{(m)}\epsilon^2}{512k^2 + 22k\epsilon}\Big\} \cdot \int_\eta^\infty (\frac{64k^2}{\epsilon})^{hd} d\epsilon \\ &\leq \eta + \gamma^{-\frac{hd}{2}} \exp\Big\{\frac{n^{(m)}\epsilon^2}{512k^2 + 22k\epsilon}\Big\} \cdot (\frac{64k^2}{\epsilon})^{hd} \eta \cdot \frac{1}{hd-1} \\ &\leq \eta + \gamma^{-\frac{hd}{2}} \exp\Big\{\frac{n^{(m)}\epsilon^2}{512k^2 + 22k\epsilon}\Big\} \cdot (n^{(m)})^{hd} \cdot \frac{\eta}{hd-1} \,. \end{aligned} \tag{2.31}$$

Setting $\eta = \gamma^{-\frac{hd}{2}} \exp\{-\frac{n^{(m)}\eta^2}{512k^2+22k\eta}\}(n^{(m)})^{hd}\frac{\eta}{hd-1}$, we get:

$$(\frac{\sqrt{\gamma}}{n^{(m)}})^{hd}(hd-1) = \exp\Big\{-\frac{n^{(m)\eta^2}}{512k^2 + 22k\eta}\Big\}. \tag{2.32}$$

From this equation, we can deduce that

$$\eta \leq \frac{khd[\log(n^{(m)}) - \log\sqrt{\gamma}]}{n^{(m)}} + 50k^2\sqrt{\frac{hd(\log n^{(m)} - \log\sqrt{\gamma})}{n^{(m)}}} \,. \tag{2.33}$$

Hence,

$$\begin{aligned} & E_{\rho^n}(E_1) \leq \; 2\eta \\ \leq & \frac{2khd(\log(n^{(m)} - \log\sqrt{\gamma})}{n^{(m)}} + 100k^2\sqrt{\frac{hd(\log(n^{(m)} - \log\sqrt{\gamma})}{n^{(m)}}} \,. \end{aligned} \tag{2.34}$$

On the other hand, the definition $f_{\mathbf{z}}$ tells us that

$$\begin{aligned}
& E_{\rho^n}(E_2) \\
&= E_{\rho^n}\Big(\inf_{f\in\mathcal{M}_h} \{\mathcal{E}_{\mathbf{z}}(f) - \mathcal{E}_{\mathbf{z}}(f_\rho) + \gamma \sum_{j=1}^{d} \left\| f^{(j)} \right\|_{\ell_2}^2 \} \Big) \\
&\leq \inf_{f\in\mathcal{M}_h} \Big\{ E_{\rho^n}(\mathcal{E}_{\mathbf{z}}(f) - \mathcal{E}_{\mathbf{z}}(f_\rho)) + \gamma \sum_{j=1}^{d} \left\| f^{(j)} \right\|_{\ell_2}^2 \Big\} \\
&\leq \inf_{f\in\mathcal{M}_h} \Big\{ \int_{\mathcal{X}} (f(x) - f_\rho(x))^2 d\rho_{\mathcal{X}}(x) + \gamma \sum_{j=1}^{d} \left\| f^{(j)} \right\|_{\ell_2}^2 \Big\}.
\end{aligned} \tag{2.35}$$

Combining Eq.(2.27) - (2.35), we get the desired result in Theorem 1. □

2.5 Experimental Results

In this section, we conduct experiments on the ADNI cohort. The goal of QTL identification is to predict brain imaging features given the SNP data. Meanwhile, we expect the model to show the importance of different SNPs, which is fundamental to understanding the role of each genetic variant in Alzheimer's disease.

2.5.1 Data Description

The data used in this work were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). One of the goals of ADNI is to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of mild cognitive impairment (MCI) and early AD. For the latest information, see `www.adni-info.org`. The genotype data [121] for all non-Hispanic Caucasian participants from the ADNI Phase 1 cohort were used here. They were genotyped using the Human 610-Quad BeadChip. Among all the SNPs, only SNPs within the boundary of $\pm 20K$ base pairs of the 153 AD candidate genes listed on the AlzGene database (`www.alzgene.org`) as of 4/18/2011 [13], were selected after the standard quality control (QC) and imputation steps. The QC criteria for the SNP

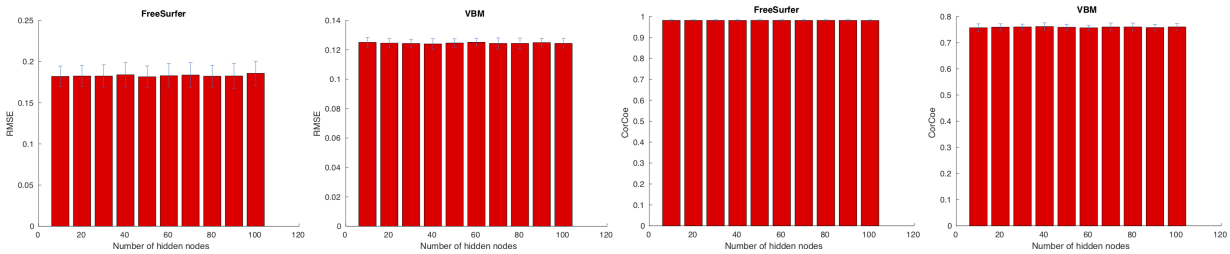

Figure 1: Performance of RMSE and CorCoe sensitivity of FNAM *w.r.t.* the number of hidden nodes.

data include (1) call rate check per subject and per SNP marker, (2) gender check, (3) sibling pair identification, (4) the Hardy-Weinberg equilibrium test, (5) marker removal by the minor allele frequency and (6) population stratification. As the second pre-processing step, the QC'ed SNPs were imputed using the MaCH software [80] to estimate the missing genotypes. As a result, our analyses included 3,123 SNPs extracted from 153 genes (boundary: $\pm 20KB$) using the ANNOVAR annotation (`http://annovar.openbioinformatics.org`).

As described previously, two widely employed automated MRI analysis techniques were used to process and extract imaging phenotypes from scans of ADNI participants [126]. First, Voxel-Based Morphometry (VBM) [6] was performed to define global gray matter (GM) density maps and extract local GM density values for 90 target regions. Second, automated parcellation via FreeSurfer V4 [44] was conducted to define volumetric and cortical thickness values for 90 regions of interest (ROIs) and to extract total intracranial volume (ICV). Further details are available in [126]. All these measures were adjusted for the baseline ICV using the regression weights derived from the healthy control (HC) participants. All 749 participants with no missing MRI measurements were included in this study, including 330 AD samples, and 210 MCI samples and 209 health control (HC) samples. In this study, we focus on a subset of these 90 imaging features which are reported to be related with AD. We extract these QTs from roughly matching regions of interest (ROIs) with VBM and FreeSurfer. Please see [147] for details. We select 26 measures for FreeSurfer, 36 measures for VBM and summarize these measures in Table 2 and Table 1.

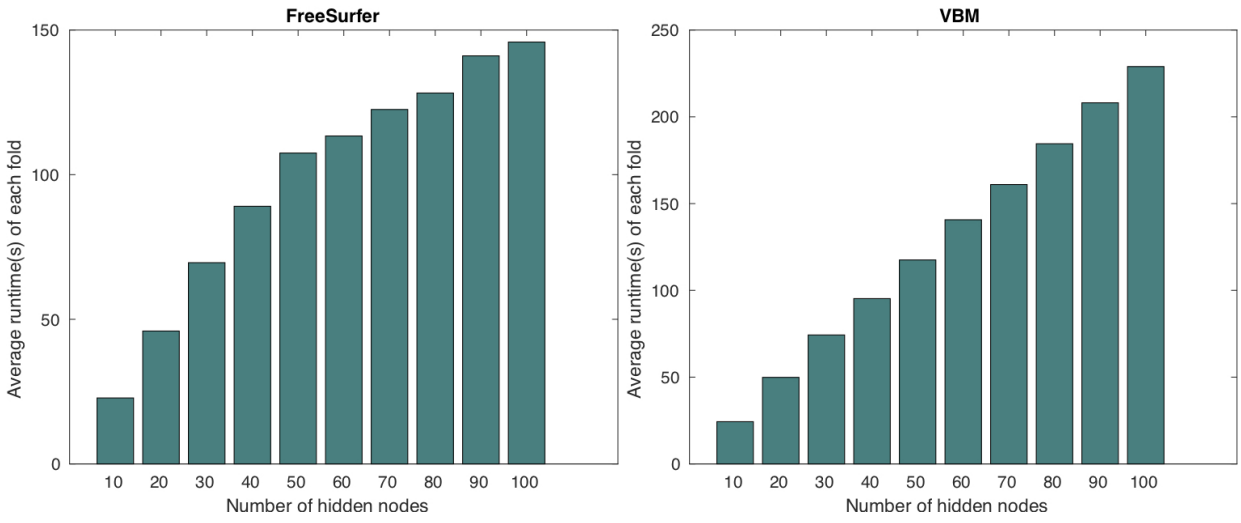

Figure 2: Runtime (in seconds) comparison of FNAM *w.r.t.* number of hidden nodes.

2.5.2 Experimental Setting

To evaluate the performance of our FNAM model, we compare with the following related methods: **LSR** (Least square regression), **RR** (Ridge regression), **Lasso** (LSR with ℓ_1-norm regularization), **Trace** (LSR with trace norm regularization), and **FNNRW** (Feedforward neural network with random weights), where we consider the Frobenius norm loss in the R_{emp} term of ([112]) for regression problem. We add a comparing method, **FNNRW-Linear** (FNNRW using linear activation function), which use linear activation function $\phi(\mathbf{x}) = \mathbf{x}$ to illustrate the contribution of the nonlinearity of activation function.

As for evaluation metric, we calculate root mean square error (RMSE) and correlation coefficient (CorCoe) between the predicted value and ground truth in out-of-sample prediction. We normalize the RMSE value via Frobenius norm of the ground truth matrix. In comparison, we adopt 5-fold cross validation and report the average performance on these 5 trials for each method.

We tune the hyper-parameter of all models in the range of $\{10^{-4},\ 10^{-3.5},\ \ldots,\ 10^{4}\}$ via nested 5-fold cross validation on the training data, and report the best parameter *w.r.t.* RMSE of each method. For methods involving feedforward neural networks, *i.e.,* FNNRW, FNNRW-Linear, and FNAM, we set $h = 50$. For FNNRW and FNAM, we set $\phi(.)$ as the tanh function.

2.5.3 Performance Comparison on ADNI Cohort

We summarize the RMSE and CorCoe comparison results in Table 3. From the results we notice that FNAM outperforms all the counterparts in both FreeSurfer and VBM. Besides, from the comparison between Lasso, Trace and FNAM, we find that the assumptions imposed by Lasso (assumption of sparse structure) and Trace (low-rank assumption) may not be appropriate when the distribution of the real data does not conform to such assumptions. In contrast, FNAM is more flexible and adaptive since FNAM does not make such structure assumption on the data distribution. Moreover, from the comparison between FNNRW, FNNRW-Linear and FNAM, we find that both FNNRW and FNAM outperform FNNRW-Linear, which demonstrates the importance of the nonlinearity introduced by the activation function. FNNRW-Linear only involves linear functions, thus is not able to show the non-linear influence of QTL. As for FNNRW, we deem that the reason for FNAM to perform better than FNNRW lies in the additive mechanism of FNAM. Since FNNRW incorporates all features in each computation, it seems too complex for the prediction thus brings about high variance.

2.5.4 Important SNP Discovery

Here we look into the significant SNPs in the prediction. According to the definition in Eq. (2.9), we calculate the importance of each SNP and select the top 10 SNPs that weigh the most in VBM analysis.

We plot the weight map and brain map of the top 10 SNPs in Figure 3. The weight matrix is calculated on the whole VBM data so as to avoid the randomness introduced by fold split. (a) Heat map showing the weights calculated via Eq. (2.9) of the top 10 SNPs in the prediction. (b) Weight matrix mapped on the brain for the VBM analysis. Different colors are employed to denote different ROIs. From the results, we notice that ApoE-rs429358 ranks the first in our prediction. As the major known genetic risk factor of AD, ApoE has been reported to be related with lowered parietal [133], temporal [141], and posterior cingulate cerebral glucose metabolism [81] of AD patients. Moreover, we present the LocusZoom plot [111] for the SNPs close to LIPA gene (10M boundary) in Chromosome 10 to show the AD-associated region around LIPA-rs885561 in Figure 4. Similar to ApoE, LIPA gene is also known to be involved in cholesterol metabolism [105],

where elevated cholesterol levels lead to higher risk of developing AD. In addition, we detect other SNPs that are established AD risk factors, *e.g.*, rs1639-PON2 [127] and rs2070045-SORL1 [116]. Replication of these results demonstrate the validity of our model.

We also pick out SNPs with potential risks whose influence on AD has not been clearly revealed in literature. For example, rs727153-LRAT is known to be related with several visual diseases, including early-onset severe retinal dystrophy and Leber congenital amaurosis 14 [109]. LRAT catalyzes the esterification of all-trans-retinol into all-trans-retinyl ester, which is essential for vitamin A metabolism in the visual system [47]. Clinically, vitamin A have been demonstrated to slow the progression of dementia and there are reports showing an trend of lower vitamin A level in AD patients [104]. Thus, it would interesting to look into the molecular role of LRAT in the progression of AD in future study. Such findings may provide insights into the discovery of new AD-associated genetic variations as well as the prevention and therapy of this disease.

2.5.5 Performance with Varying Hidden Node Number

In Algorithm 1, we need to predefine the number of hidden nodes h, thus it is crucial to test if the performance of FNAM is stable with different h. In this section, we analyze the stability of FNAM model *w.r.t.* the choice of hidden node number. Figure 1 display the RMSE and CorCoe comparison results of FNAM when h is set in the range of $\{10,\ 20,\ \ldots,\ 100\}$. From these results we can find that our FNAM model performs quite stable *w.r.t.* the choice of hidden node number. As a consequence, we do not need to make much effort on tuning the number of hidden nodes. This is important to an efficient implementation in practice.

2.5.6 Running Time Analysis

Here we present experimental results to analyze the runtime (in seconds) of FNAM with different number of hidden nodes. Our experiments are conducted on a 24-core Intel(R) Xeon(R) E5-2620 v3 CPU @ 2.40GHz server with 65GB memory. The operating system is Ubuntu 16.04.1 and the software we use is Matlab R2016a (64-bit) 9.0.0. Seen from Figure 2, it only takes a few minutes to run our model on the ADNI data. Y-axis shows the average runtime of one fold in the cross validation, including the time for tuning hyperparameter γ as well as the time for obtaining

the prediction results. The running time is roughly linear to the number of hidden nodes, which is consistent with our theoretical analysis that the time complexity of FNAM is $O(ndhc)$. This result further illustrates the efficiency of our model, such that we can use the model in larger scale case in an efficient manner.

Table 1: 36 GM density measures (VBM) matched with disease-related ROIs.

GM Density ID	**ROI**
LHippocampus, RHippocampus	Hippocampus
LParahipp, RParahipp	Parahippocampal gyrus
LPrecuneus, RPrecuneus	Precuneus
LInfFrontal_Oper, RInfFrontal_Oper	Inferior frontal operculum
LInfOrbFrontal, RInfOrbFrontal	Inferior orbital frontal gyrus
LInfFrontal_Triang, RInfFrontal_Triang	Inferior frontal triangularis
LMedOrbFrontal, RMedOrbFrontal	Medial orbital frontal gyrus
LMidFrontal, RMidFrontal	Middle frontal gyrus
LMidOrbFrontal, RMidOrbFrontal	Middle orbital frontal gyrus
LSupFrontal, RSupFrontal	Superior frontal gyrus
LMedSupFrontal, RMedSupFrontal	Medial superior frontal gyrus
LSupOrbFrontal, RSupOrbFrontal	Superior orbital frontal gyrus
LRectus, RRectus	Rectus gyrus
LRolandic_Oper, RRolandic_Oper	Rolandic operculum
LSuppMotorArea, RSuppMotorArea	Supplementary motor area
LInfTemporal, RInfTemporal	Inferior temporal gyrus
LMidTemporal, RMidTemporal	Middle temporal gyrus
LSupTemporal, RSupTemporal	Superior temporal gyrus

Table 2: 26 volumetric/thickness measures (FreeSurfer) matched with disease-related ROIs.

Volume/Thickness ID	**ROI**
LHippVol, RHippVol	Volume of hippocampus
LEntCtx, REntCtx	Thickness of entorhinal cortex
LParahipp, RParahipp	Thickness of parahippocampal gyrus
LPrecuneus, RPrecuneus	Thickness of precuneus
LCaudMidFrontal, RCaudMidFrontal	Mean thickness of caudal midfrontal
LRostMidFrontal, RRostMidFrontal	Mean thickness of rostral midfrontal
LSupFrontal, RSupFrontal	Mean thickness of superior frontal
LLatOrbFrontal, RLatOrbFrontal	Mean thickness of lateral orbitofrontal
LMedOrbFrontal, RMedOrbFrontal	Mean thickness of medial orbitofrontal gyri
LFrontalPole, RFrontalPole	Mean thickness of frontal pole
LInfTemporal, RInfTemporal	Mean thickness of inferior temporal
LMidTemporal, RMidTemporal	Mean thickness of middle temporal
LSupTemporal, RSupTemporal	Mean thickness of superior temporal gyri

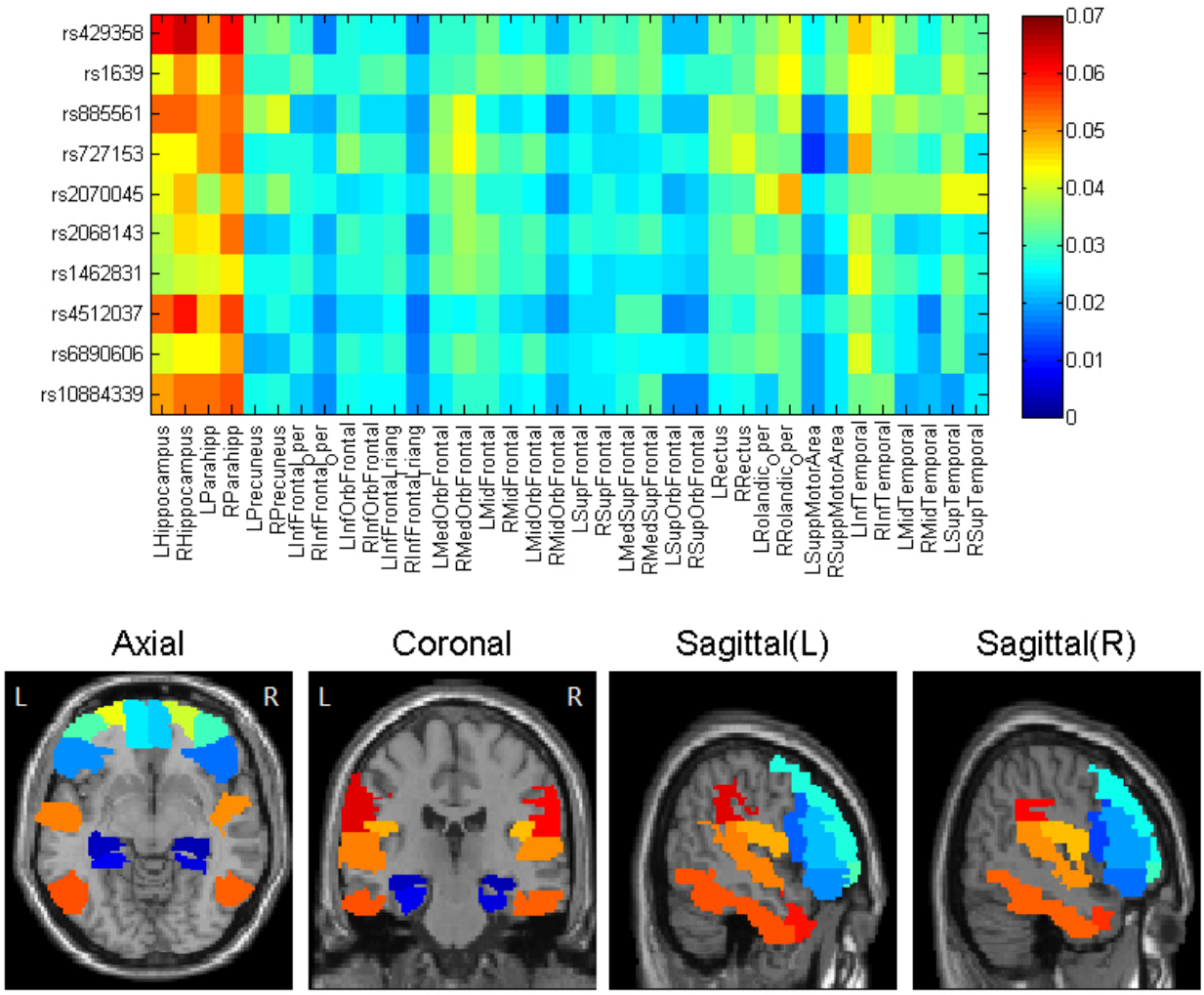

Figure 3: Heat map (upper) and brain map (below) of the top 10 SNPs identified by FNAM in VBM analysis.

Table 3: Performance evaluation of FNAM on FreeSurfer and VBM prediction.

		FreeSurfer	**VBM**
RMSE	**LSR**	0.258±0.012	0.175±0.005
	RR	0.184±0.013	0.129±0.004
	Lasso	0.253±0.012	0.128±0.004
	Trace	0.197±0.015	0.139±0.005
	FNNRW-Linear	0.244±0.019	0.199±0.017
	FNNRW	0.227±0.022	0.168±0.027
	FNAM	**0.182±0.013**	**0.125±0.003**
CorCoe	**LSR**	0.965±0.003	0.585±0.019
	RR	0.982±0.003	0.744±0.014
	Lasso	0.966±0.003	0.748±0.014
	Trace	0.979±0.003	0.710±0.019
	FNNRW-Linear	0.968±0.004	0.512±0.059
	FNNRW	0.972±0.006	0.604±0.082
	FNAM	**0.982±0.003**	**0.759±0.011**

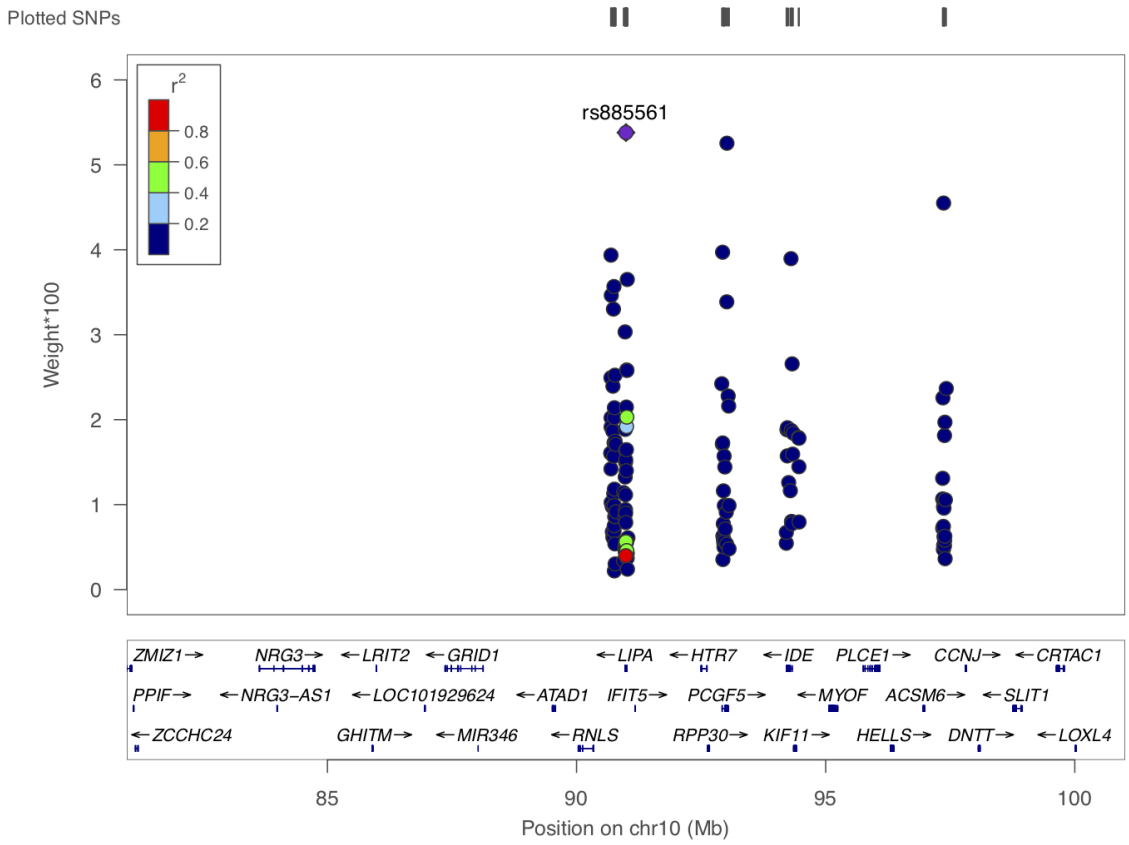

Figure 4: LocusZoom plot showing Alzheimer's associated region around rs885561-LIPA (10M boundary) in Chromosome 10.

3.0 Uncovering Feature Group via Structured Additive Model

3.1 Introduction

The additive models based on statistical learning methods have been playing important roles for the high-dimensional data analysis due to their well performance on prediction tasks and variable selection (deep learning models often don't work well when the number of training data is not large). In essential, additive models inherit the representation flexibility of nonlinear models and the interpretability of linear models. For a learning approach under additive models, there are two key components: the hypothesis function space and the regularizer to address certain restrictions on estimator. Different from traditional learning methods, the hypothesis space used in additive models is relied on the decomposition of input vector. Usually, each input vector $X \in \mathbb{R}^p$ is divided into p parts directly [114, 173, 23, 167] or some subgroups according to prior structural information among input variables [166, 165]. The component function is defined on each decomposed input and the hypothesis function is constructed by the sum of all component functions. Typical examples of hypothesis space include the kernel-based function space [113, 23, 64] and the spline-based function space [83, 94, 58, 173]. Moreover, the Tikhonov regularization scheme has been used extensively for constructing the additive models, where the regularizer is employed to control the complexity of hypothesis space. The examples of regularizer include the kernel-norm regularization associated with the reproducing kernel Hilbert space (RKHS) [22, 23, 64] and various sparse regularization [114, 173, 165].

More recently several group sparse additive models have been proposed to tackle the high-dimensional regression problem due to their nice theoretical properties and empirical effectiveness [94, 58, 165]. However, most existing additive model based learning approaches are mainly limited to the least squares regression problem and spline-based hypothesis spaces. Surprisingly, there is no any algorithmic design and theoretical analysis for classification problem with group sparse additive models in RKHS. This chapter focuses on filling in this gap on algorithmic design and learning theory for additive models. A novel sparse classification algorithm, called as *group sparse additive machine* (GroupSAM), is proposed under a coefficient-based regularized frame-

work, which is connected to the linear programming support vector machine (LPSVM) [142, 159]. By incorporating the grouped variables with prior structural information and the $\ell_{2,1}$-norm based structured sparse regularizer, the new GroupSAM model can conduct the nonlinear classification and variable selection simultaneously. Similar to the sparse additive machine (SAM) in [173], our GroupSAM model can be efficiently solved via proximal gradient descent algorithm. The main contributions of this chapter can summarized in two-fold:

- A new group sparse nonlinear classification algorithm (GroupSAM) is proposed by extending the previous additive regression models to the classification setting, which contains the LPSVM with additive kernel as its special setting. To the best of our knowledge, this is the first algorithmic exploration of additive classification models with group sparsity.
- Theoretical analysis and empirical evaluations on generalization ability are presented to support the effectiveness of GroupSAM. Based on constructive analysis on the hypothesis error, we get the estimate on the excess generalization error, which shows that our GroupSAM model can achieve the fast convergence rate $O(n^{-1})$ under mild conditions. Experimental results demonstrate the competitive performance of GroupSAM over the related methods on both simulated and real data.

Before ending this section, we discuss related works. In [22], support vector machine (SVM) with additive kernels was proposed and its classification consistency was established. Although this method can also be used for grouped variables, it only focuses on the kernel-norm regularizer without addressing the sparseness for variable selection. In [173], the SAM was proposed to deal with the sparse representation on the orthogonal basis of hypothesis space. Despite good computation and generalization performance, SAM does not explore the structure information of input variables and ignores the interactions among variables. More important, different from finite splines approximation in [173], our approach enables us to estimate each component function directly in RKHS. As illustrated in [138, 88], the RKHS-based method is flexible and only depends on few tuning parameters, but the commonly used spline methods need specify the number of basis functions and the sequence of knots.

It should be noticed that the group sparse additive models (GroupSpAM in [165]) also address the sparsity on the grouped variables. However, there are key differences between GroupSAM

Table 4: Properties of different additive models.

	SAM [173]	Group Lasso[166]	GroupSpAM [165]	GroupSAM
Hypothesis space	data-independent	data-independent	data-independent	data-dependent
Loss function	hinge loss	least-square	least-square	hinge loss
Group sparsity	No	Yes	Yes	Yes
Generalization bound	Yes	No	No	Yes

and GroupSpAM: *1) Hypothesis space.* The component functions in our model are obtained by searching in kernel-based data dependent hypothesis spaces, but the method in [165] uses data independent hypothesis space (not associated with kernel). As shown in [129, 128, 19, 161], the data dependent hypothesis space can provide much more adaptivity and flexibility for nonlinear prediction. The advantage of kernel-based hypothesis space for additive models is also discussed in [88]. *2) Loss function.* The hinge loss used in our classification model is different from the least-squares loss in [165]. *3) Optimization.* Our GroupSAM only needs to construct one component function for each variable group, but the model in [165] needs to find the component functions for each variable in a group. Thus, our method is usually more efficient. Due to the kernel-based component function and non-smooth hinge loss, the optimization of GroupSpAM can not be extended to our model directly. *4) Learning theory.* We establish the generalization bound of GroupSAM by the error estimate technique with data dependent hypothesis spaces, while the error bound is not covered in [165]. Now, we present a brief summary in Table 4 to better illustrate the differences of our GroupSAM with other methods.

3.2 Group Sparse Additive Machine

In this section, we first revisit the basic background of binary classification and additive models, and then introduce our new GroupSAM model.

Let $\mathcal{Z} := (\mathcal{X}, \mathcal{Y}) \subset \mathbb{R}^{p+1}$, where $\mathcal{X} \subset \mathbb{R}^p$ is a compact input space and $\mathcal{Y} = \{-1, 1\}$ is the set of labels. We assume that the training samples $\mathbf{z} := \{z_i\}_{i=1}^n = \{(x_i, y_i)\}_{i=1}^n$ are independently drawn from an unknown distribution ρ on $\mathcal{Z}$, where each $x_i \in \mathcal{X}$ and $y_i \in \{-1, 1\}$. Let's denote the marginal distribution of ρ on $\mathcal{X}$ as $\rho_{\mathcal{X}}$ and denote its conditional distribution for given $x \in \mathcal{X}$ as $\rho(\cdot|x)$.

For a real-valued function $f : \mathcal{X} \to \mathbb{R}$, we define its induced classifier as $\operatorname{sgn}(f)$, where $\operatorname{sgn}(f)(x) = 1$ if $f(x) \geq 0$ and $\operatorname{sgn}(f)(x) = -1$ if $f(x) < 0$. The prediction performance of f is measured by the misclassification error:

$$\mathcal{R}(f) = \operatorname{Prob}\{Yf(X) \leq 0\} = \int_{\mathcal{X}} \operatorname{Prob}(Y \neq \operatorname{sgn}(f)(x)|x) d\rho_{\mathcal{X}}. \tag{3.1}$$

It is well known that the minimizer of $\mathcal{R}(f)$ is the Bayes rule:

$$f_c(x) = \operatorname{sgn}\Big(\int_{\mathcal{Y}} y d\rho(y|x)\Big) = \operatorname{sgn}\Big(\operatorname{Prob}(y = 1|x) - \operatorname{Prob}(y = -1|x)\Big). \tag{3.2}$$

Since the Bayes rule involves the unknown distribution ρ, it can not be computed directly. In machine learning literature, the classification algorithm usually aims to find a good approximation of f_c by minimizing the empirical misclassification risk:

$$\mathcal{R}_{\mathbf{z}}(f) = \frac{1}{n} \sum_{i=1}^{n} I(y_i f(x_i) \leq 0)\,, \tag{3.3}$$

where $I(A) = 1$ if A is true and 0 otherwise. However, the minimization problem associated with $\mathcal{R}_{\mathbf{z}}(f)$ is NP-hard due to the $0-1$ loss I. To alleviate the computational difficulty, various convex losses have been introduced to replace the $0-1$ loss, *e.g.*, the hinge loss, the least square loss, and the exponential loss [171, 9, 29]. Among them, the hinge loss is the most popular error metric for classification problem due to its nice theoretical properties. In this chapter, following [22, 173], we use the hinge loss $\ell(y, f(x)) = (1-yf(x))_+ = \max\{1-yf(x), 0\}$to measure the misclassification cost.The expected and empirical risks associated with the hinge loss are defined respectively as:

$$\mathcal{E}(f) = \int_{\mathcal{Z}} (1 - yf(x))_+ d\rho(x, y)\,, \tag{3.4}$$

and

$$\mathcal{E}_{\mathbf{z}}(f) = \frac{1}{n} \sum_{i=1}^{n} (1 - y_i f(x_i))_+\,. \tag{3.5}$$

In theory, the excess misclassification error $\mathcal{R}(\mathrm{sgn}(f)) - \mathcal{R}(f_c)$ can be bounded by the excess convex risk $\mathcal{E}(f) - \mathcal{E}(f_c)$ [171, 9, 29]. Therefore, the classification algorithm usually is constructed under structural risk minimization [142] associated with $\mathcal{E}_{\mathbf{z}}(f)$.

In this chapter, we propose a novel group sparse additive machine (GroupSAM) for nonlinear classification. Let $\{1, \cdots, p\}$ be partitioned into d groups. For each $j \in \{1, ..., d\}$, we set $\mathcal{X}^{(j)}$ as the grouped input space and denote $f^{(j)} : \mathcal{X}^{(j)} \to \mathbb{R}$ as the corresponding component function. Usually, the groups can be obtained by prior knowledge [165] or be explored by considering the combinations of input variables [64].

Let each $K^{(j)} : \mathcal{X}^{(j)} \times \mathcal{X}^{(j)} \to \mathbb{R}$ be a Mercer kernel and let $\mathcal{H}_{K^{(j)}}$ be the corresponding RKHS with norm $\|\cdot\|_{K^{(j)}}$. It has been proved in [22] that

$$\mathcal{H} = \Big\{ \sum_{j=1}^{d} f^{(j)} : f^{(j)} \in \mathcal{H}_{K^{(j)}}, 1 \le j \le d \Big\} \tag{3.6}$$

with norm

$$\|f\|_K^2 = \inf \Big\{ \sum_{j=1}^{d} \|f^{(j)}\|_{K^{(j)}}^2 : f = \sum_{j=1}^{d} f^{(j)} \Big\} \tag{3.7}$$

is an RKHS associated with the additive kernel $K = \sum_{j=1}^{d} K^{(j)}$.

For any given training set $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^n$, the additive model in $\mathcal{H}$ can be formulated as:

$$\bar{f}_{\mathbf{z}} = \underset{f=\sum_{j=1}^{d} f^{(j)} \in \mathcal{H}}{\arg\min} \Big\{ \mathcal{E}_{\mathbf{z}}(f) + \eta \sum_{j=1}^{d} \tau_j \|f^{(j)}\|_{K^{(j)}}^2 \Big\}, \tag{3.8}$$

where $\eta = \eta(n)$ is a positive regularization parameter and $\{\tau_j\}$ are positive bounded weights for different variable groups.

The solution $\bar{f}_{\mathbf{z}}$ in (3.8) has the following representation:

$$\bar{f}_{\mathbf{z}}(x) = \sum_{j=1}^{d} \bar{f}_{\mathbf{z}}{}^{(j)}(x^{(j)}) = \sum_{j=1}^{d} \sum_{i=1}^{n} \bar{\alpha}_{\mathbf{z},i}^{(j)} y_i K^{(j)}(x_i^{(j)}, x^{(j)}),\ \bar{\alpha}_{\mathbf{z},i}^{(j)} \in \mathbb{R},\ 1 \le i \le n,\ 1 \le j \le d\,. \tag{3.9}$$

Observe that $\bar{f}_{\mathbf{z}}{}^{(j)}(x) \equiv 0$ is equivalent to $\bar{\alpha}_{\mathbf{z},i}^{(j)} = 0$ for all i. Hence, we expect $\|\bar{\alpha}_{\mathbf{z}}^{(j)}\|_2 = 0$ for $\bar{\alpha_{\mathbf{z}}}^{(j)} = (\bar{\alpha}_{\mathbf{z},1}^{(j)}, \cdots, \bar{\alpha}_{\mathbf{z},n}^{(j)})^T \in \mathbb{R}^n$ if the j-th variable group is not truly informative. This motivation pushes us to consider the sparsity-induced penalty:

$$\Omega(f) = \inf \Big\{ \sum_{j=1}^{d} \tau_j \|\alpha^{(j)}\|_2 : f = \sum_{j=1}^{d} \sum_{i=1}^{n} \alpha_i^{(j)} y_i K^{(j)}(x_i^{(j)}, \cdot) \Big\}\,. \tag{3.10}$$

This group sparse penalty aims at the variable selection [166] and was introduced into the additive regression model [165].

Inspired by learning with data dependent hypothesis spaces [129], we introduce the following hypothesis spaces associated with training samples $\mathbf{z}$:

$$\mathcal{H}_{\mathbf{z}} = \Big\{ f = \sum_{j=1}^{d} f^{(j)} : f^{(j)} \in \mathcal{H}_{\mathbf{z}}^{(j)} \Big\}, \tag{3.11}$$

where

$$\mathcal{H}_{\mathbf{z}}^{(j)} = \Big\{ f^{(j)} = \sum_{i=1}^{n} \alpha_i^{(j)} K^{(j)}(x_i^{(j)}, \cdot) : \alpha_i^{(j)} \in \mathbb{R} \Big\}. \tag{3.12}$$

Under the group sparse penalty and data dependent hypothesis space, the group sparse additive machine (GroupSAM) can be written as:

$$f_{\mathbf{z}} = \arg\min_{f \in \mathcal{H}_{\mathbf{z}}} \Big\{ \frac{1}{n} \sum_{i=1}^{n} (1 - y_i f(x_i))_+ + \lambda \Omega(f) \Big\}, \tag{3.13}$$

where $\lambda > 0$ is a regularization parameter.

Let's denote $\alpha^{(j)} = (\alpha_1^{(j)}, \cdots, \alpha_n^{(j)})^T$ and $\mathbf{K}_i^{(j)} = (K^{(j)}(x_1^{(j)}, x_i^{(j)}), \cdots, K^{(j)}(x_n^{(j)}, x_i^{(j)}))^T$. The GroupSAM in (3.13) can be rewritten as:

$$f_{\mathbf{z}} = \sum_{j=1}^{d} f_{\mathbf{z}}^{(j)} = \sum_{j=1}^{d} \sum_{t=1}^{n} \alpha_{\mathbf{z},t}^{(j)} K^{(j)}(x_t^{(j)}, \cdot)\,, \tag{3.14}$$

with $\{\alpha_{\mathbf{z}}^{(j)}\} = \arg\min_{\alpha^{(j)} \in \mathbb{R}^n, 1 \le j \le d} \Big\{ \frac{1}{n} \sum_{i=1}^{n} \big(1 - y_i \sum_{j=1}^{d} (\mathbf{K}_i^{(j)})^T \alpha^{(j)}\big)_+ + \lambda \sum_{j=1}^{d} \tau_j \|\alpha^{(j)}\|_2 \Big\}$.

The above formulation transforms the function-based learning problem (3.13) into a coefficient-based learning problem in a finite dimensional vector space. The solution of (3.13) is spanned naturally by the kernelized functions $\{K^{(j)}(\cdot, x_i^{(j)}))\}$, rather than B-Spline basis functions [173]. When $d = 1$, our GroupSAM model degenerates to the special case which includes the LPSVM loss and the sparsity regularization term. Compared with LPSVM [142, 159] and SVM with additive kernels [22], our GroupSAM model imposes the sparsity on variable groups to improve the prediction interpretation of additive classification model.

For given $\{\tau_j\}$, the optimization problem of GroupSAM can be computed efficiently via an accelerated proximal gradient descent algorithm developed in [173]. Due to space limitation, we don't recall the optimization algorithm here again.

3.3 Generalization Error Bound

In this section, we will derive the estimate on the excess misclassification error $\mathcal{R}(\mathrm{sgn}(f_{\mathbf{z}})) - \mathcal{R}(f_c)$. Before providing the main theoretical result, we introduce some necessary assumptions for learning theory analysis.

Assumption A. The intrinsic distribution ρ on $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ satisfies the Tsybakov noise condition with exponent $0 \leq q \leq \infty$. That is to say, for some $q \in [0, \infty)$ and $\Delta > 0$,

$$\rho_{\mathcal{X}}\Big(\{x \in \mathcal{X} : |\mathrm{Prob}(y = 1|x) - \mathrm{Prob}(y = -1|x)| \leq \Delta t\}\Big) \leq t^q, \forall t > 0. \tag{3.15}$$

The Tsybakov noise condition was proposed in [140] and has been used extensively for theoretical analysis of classification algorithms [159, 29, 158, 138]. Indeed, (3.15) holds with exponent $q = 0$ for any distribution and with $q = \infty$ for well separated classes. Next we introduce the empirical covering numbers [37] to measure the capacity of hypothesis space.

Definition 1. *Let $\mathcal{F}$ be a set of functions on $\mathcal{Z}$ with $\mathbf{u} = \{u_i\}_{i=1}^k \subset \mathcal{Z}$. Define the ℓ_2-empirical metric as $\ell_{2,\mathbf{u}}(f, g) = \big\{\frac{1}{n}\sum_{t=1}^k (f(u_t) - g(u_t))^2\big\}^{\frac{1}{2}}$. The covering number of $\mathcal{F}$ with ℓ_2-empirical metric is defined as $\mathcal{N}_2(\mathcal{F}, \varepsilon) = \sup_{n\in\mathbb{N}} \sup_{\mathbf{u}\in\mathcal{X}^n} \mathcal{N}_{2,\mathbf{u}}(\mathcal{F}, \varepsilon)$, where*

$$\mathcal{N}_{2,\mathbf{u}}(\mathcal{F}, \varepsilon) = \inf\Big\{l \in \mathbb{N} : \exists\{f_i\}_{i=1}^l \subset \mathcal{F} \ \ s.t. \ \ \mathcal{F} = \bigcup_{i=1}^l \{f \in \mathcal{F} : \ell_{2,\mathbf{u}}(f, f_i) \leq \varepsilon\}\Big\}. \tag{3.16}$$

Let $\mathcal{B}_r = \{f \in \mathcal{H}_K : \|f\|_K \leq r\}$ and $\mathcal{B}_r^{(j)} = \{f^{(j)} \in \mathcal{H}_{K^{(j)}} : \|f^{(j)}\|_{K^{(j)}} \leq r\}$.

Assumption B. Assume that $\kappa = \sum_{j=1}^d \sup_{x^{(j)}} \sqrt{K^{(j)}(x^{(j)}, x^{(j)})} < \infty$ and for some $s \in (0, 2), c_s > 0$,

$$\log \mathcal{N}_2(\mathcal{B}_1^{(j)}, \varepsilon) \leq c_s \varepsilon^{-s}, \ \forall \varepsilon > 0, \ j \in \{1, ..., d\}. \tag{3.17}$$

It has been asserted in [23] that under Assumption B the following holds:

$$\log \mathcal{N}_2(\mathcal{B}_1, \varepsilon) \leq c_s d^{1+s} \varepsilon^{-s}, \ \forall \varepsilon > 0. \tag{3.18}$$

It is worthy noticing that the empirical covering number has been studied extensively in learning theory literatures [37, 138]. We refer interested readers to the detailed examples provided in Theorem 2 of [129], Lemma 3 of [128], and Examples 1, 2 of [52]. The capacity condition

of additive assumption space just depends on the dimension of subspace $\mathcal{X}^{(j)}$. When $K^{(j)} \in C^{\nu}(\mathcal{X}^{(j)} \times \mathcal{X}^{(j)})$ for every $j \in \{1, \cdots, d\}$, the theoretical analysis in [129] assures that Assumption B holds true for:

$$s = \begin{cases} \frac{2d_0}{d_0+2\nu}, & \nu \in (0, 1]; \\ \frac{2d_0}{d_0+\nu}, & \nu \in [1, 1 + d_0/2]; \\ \frac{d_0}{\nu}, & \nu \in (1 + d_0/2, \infty). \end{cases} \tag{3.19}$$

Here d_0 denotes the maximum dimension among $\{\mathcal{X}^{(j)}\}$.

With respect to (3.8), we introduce the data-free regularized function f_η defined by:

$$f_\eta = \underset{f=\sum_{j=1}^d f^{(j)} \in \mathcal{H}}{\arg\min} \Big\{ \mathcal{E}(f) + \eta \sum_{j=1}^d \tau_j \|f^{(j)}\|^2_{K^{(j)}} \Big\}. \tag{3.20}$$

Inspired by the analysis in [23], we define the following as the approximation error, which reflects the learning ability of hypothesis space $\mathcal{H}$ under the Tikhonov regularization scheme.

$$D(\eta) = \mathcal{E}(f_\eta) - \mathcal{E}(f_c) + \eta \sum_{j=1}^d \tau_j \|f_\eta^{(j)}\|^2_{K^{(j)}} \tag{3.21}$$

The following approximation condition has been studied and used extensively for classification problems, such as [18, 29, 159, 158]. Please see Examples 3 and 4 in [18] for the explicit version for Soblov kernel and Gaussian kernel induced reproducing kernel Hilbert space.

Assumption C. There exists an exponent $\beta \in (0, 1)$ and a positive constant c_β such that:

$$D(\eta) \leq c_\beta \eta^\beta, \forall \eta > 0. \tag{3.22}$$

Now we introduce our main theoretical result on the generalization bound as follows.

Theorem 2. *Let* $0 < \min_j \tau_j \leq \max_j \tau_j \leq c_0 < \infty$ *and Assumptions A-C hold true. Take* $\lambda = n^{-\theta}$ *in (3.13) for* $0 < \theta \leq \min\{\frac{2-s}{2s}, \frac{3+5\beta}{2-2\beta}\}$. *For any* $\delta \in (0, 1)$, *there exists a constant* C *independent of* n, δ *such that*

$$\mathcal{R}(\mathrm{sgn}(f_{\mathbf{z}})) - \mathcal{R}(f_c) \leq C \log(3/\delta) n^{-\vartheta} \tag{3.23}$$

with confidence $1 - \delta$, *where*

$$\vartheta = \min\Big\{ \frac{q+1}{q+2}, \frac{\beta(2\theta+1)}{2\beta+2}, \frac{(q+1)(2-s-2s\theta)}{4+2q+sq}, \frac{3+5\beta+2\beta\theta-2\theta}{4+4\beta} \Big\}. \tag{3.24}$$

Theorem 2 demonstrates that GroupSAM in (3.13) can achieve the convergence rate with polynomial decay under mild conditions in hypothesis function space. When $q \to \infty$, $\beta \to 1$, and each $K^{(j)} \in C^{\infty}$, the error decay rate of GroupSAM can arbitrarily close to $O(n^{-\min\{1, \frac{1+2\theta}{4}\}})$. Hence, the fast convergence rate $O(n^{-1})$ can be obtained under proper selections on parameters. To verify the optimal bound, we need provide the lower bound for the excess misclassification error. This is beyond the main focus of this chapter and we leave it for future study.

Additionally, the consistency of GroupSAM can be guaranteed with the increasing number of training samples.

Corollary 1. *Under conditions in Theorem 2, there holds* $\mathcal{R}(\mathrm{sgn}(f_{\mathbf{z}})) - \mathcal{R}(f_c) \to 0$ *as* $n \to \infty$.

To better understand our theoretical result, we compare it with the related works as below:

1) Compared with group sparse additive models. Although the asymptotic theory of group sparse additive models has been well studied in [94, 58, 165], all of them only consider the regression task under the mean square error criterion and basis function expansion. Due to the kernel-based component function and non-smooth hinge loss, the previous analysis cannot be extended to GroupSAM directly.

2) Compared with classification with additive models. In [173], the convergence rate is presented for sparse additive machine (SAM), where the input space $\mathcal{X}$ is divided into p subspaces directly without considering the interactions among variables. Different to the sparsity on variable groups in this chapter, SAM is based on the sparse representation of orthonormal basis similar with [94]. In [22], the consistency of SVM with additive kernel is established, where the kernel-norm regularizer is used. However, the sparsity on variables and the learning rate are not investigated in previous articles.

3) Compared with the related analysis techniques. While the analysis technique used here is inspired from [159, 158], it is the first exploration for additive classification model with group sparsity. In particular, the hypothesis error analysis develops the stepping stone technique from the ℓ_1-norm regularizer to the group sparse $\ell_{2,1}$-norm regularizer. Our analysis technique also can be applied to other additive models. For example, we can extend the shrunk additive regression model in [64] to the sparse classification setting and investigate its generalization bound by the current technique.

Proof sketches of Theorem 1

To get tight error estimate, we introduce the clipping operator

$$\pi(f)(x) = \max\{-1, \min\{f(x), 1\}\},$$

which has been widely used in learning theory literatures, such as [29, 138, 159, 158]. Since $\mathcal{R}(\mathrm{sgn}(f_{\mathbf{z}})) - \mathcal{R}(f_c)$ can be bounded by $\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_c)$, we focus on bounding the excess convex risk.

Using f_η as the intermediate function, we can obtain the following error decomposition.

Proposition 1. *For $f_{\mathbf{z}}$ defined in (3.13), there holds*

$$\mathcal{R}(\mathrm{sgn}(f_{\mathbf{z}})) - \mathcal{R}(f_c) \leq \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_c) \leq E_1 + E_2 + E_3 + D(\eta), \tag{3.25}$$

where $D(\eta)$ is defined in (3.21),

$$E_1 = \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_c) - \big(\mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) - \mathcal{E}_{\mathbf{z}}(f_c)\big), \tag{3.26}$$

$$E_2 = \mathcal{E}_{\mathbf{z}}(f_\eta) - \mathcal{E}_{\mathbf{z}}(f_c) - \big(\mathcal{E}_{\mathbf{z}}(f_\eta) - \mathcal{E}(f_c)\big), \tag{3.27}$$

and

$$E_3 = \mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) + \lambda\Omega(f_{\mathbf{z}}) - \big(\mathcal{E}_{\mathbf{z}}(f_\eta) + \eta\sum_{j=1}^{d}\tau_j\|f_\eta^{(j)}\|_{K^{(j)}}^2\big). \tag{3.28}$$

In learning theory literature, $E_1 + E_2$ is called as the sample error and E_3 is named as the hypothesis error. Detailed proofs for these error terms are provided in the supplementary materials.

The upper bound of hypothesis error demonstrates that the divergence induced from regularization and hypothesis space tends to zero as $n \to \infty$ under proper selected parameters. To estimate the hypothesis error E_3, we choose $\bar{f}_{\mathbf{z}}$ as the stepping stone function to bridge $\mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) + \lambda\Omega(f_{\mathbf{z}})$ and $\mathcal{E}_{\mathbf{z}}(f_\eta) + \lambda\sum_{j=1}^{d}\tau_j\|f_\eta^{(j)}\|_{K^{(j)}}^2$. The proof is inspired from the stepping stone technique for support vector machine classification [159]. Notice that our analysis is associated with the $\ell_{2,1}$-norm regularizer while the previous analysis just focuses on the ℓ_1-norm regularization.

The error term E_1 reflects the divergence between the expected excess risk $\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_c)$ and the empirical excess risk $\mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) - \mathcal{E}_{\mathbf{z}}(f_c)$. Since $f_{\mathbf{z}}$ involves any given $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^n$, we introduce the concentration inequality in [158] to bound E_1. We also bound the error term E_2 in terms of the one-side Bernstein inequality [29].

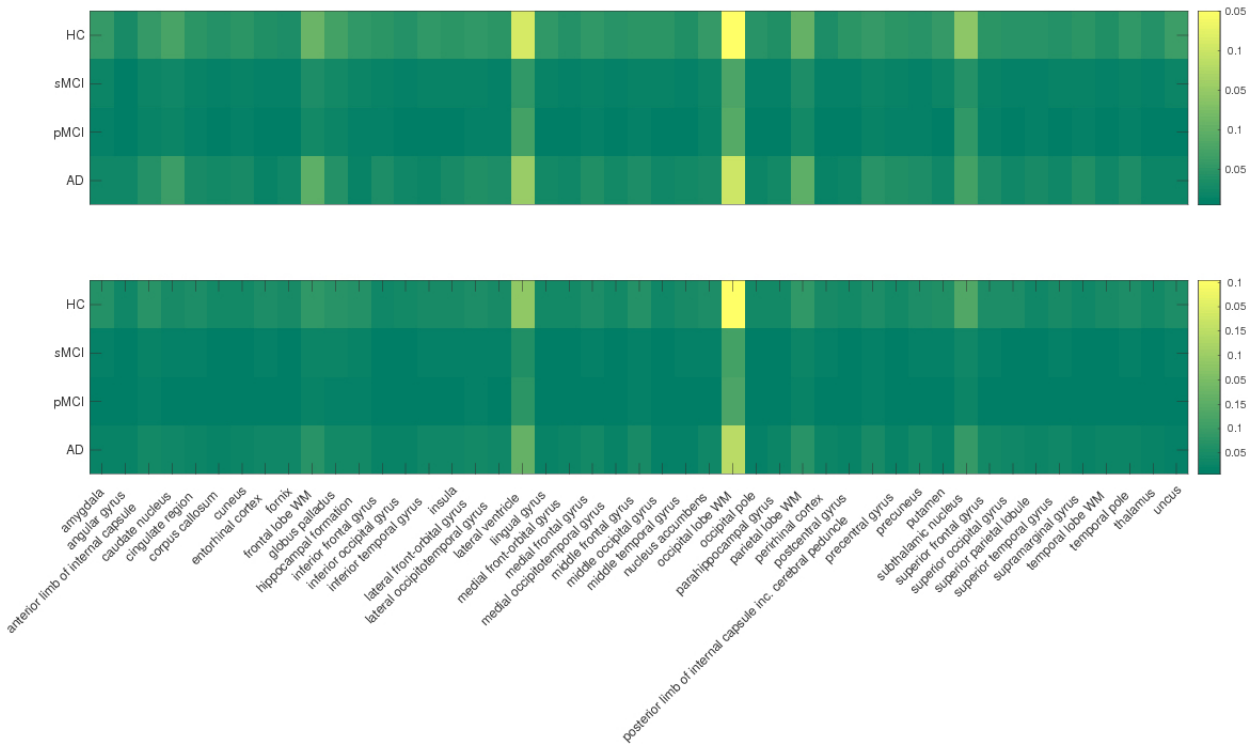

Figure 5: Heat maps of the weight matrices learned by GroupSAM on MRI data. The upper figure shows left hemisphere and the lower shows the right hemisphere.

3.4 Experimental Results

To evaluate the performance of our proposed GroupSAM model, we compare our model with the following methods: SVM (linear SVM with ℓ_2-norm regularization), L1SVM (linear SVM with ℓ_1-norm regularization), GaussianSVM (nonlinear SVM using Gaussian kernel), SAM (Sparse Additive Machine) [173], and GroupSpAM (Group Sparse Additive Models) [165] which is adapted to the classification setting.

As for evaluation metric, we calculate the classification accuracy, *i.e.,* percentage of correctly labeled samples in the prediction. In comparison, we adopt 2-fold cross validation and report the average performance of each method.

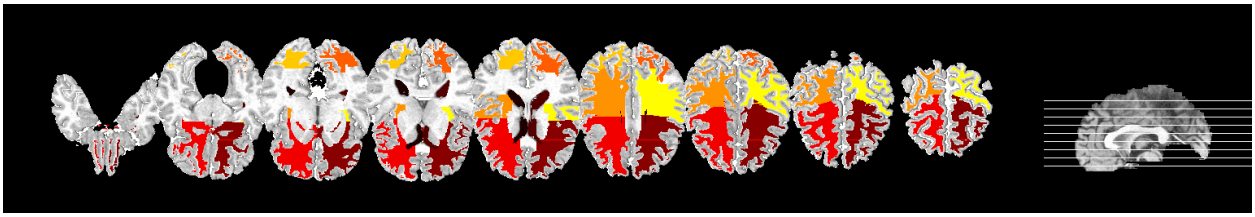

Figure 6: Cortical maps of the top 10 MRI imaging markers identified by GroupSAM.

We implement SVM, L1SVM and GaussianSVM using the LIBSVM toolbox [17]. We determine the hyper-parameter of all models, *i.e.,* parameter C of SVM, L1SVM and GaussianSVM, parameter λ of SAM, parameter λ of GroupSpAM, parameter λ of GroupSAM, in the range of $\{10^{-3},\ 10^{-2},\ \ldots,\ 10^{3}\}$. We tune the hyper-parameters via 2-fold cross validation on the training data and report the best parameter *w.r.t.* classification accuracy of each method. In the accelerated proximal gradient descent algorithm for both SAM and GroupSAM, we set $\mu = 0.5$, and the number of maximum iterations as 2000.

3.4.1 Performance Comparison on Synthetic Data

We first examine the classification performance on the synthetic data as a sanity check. Our synthetic data is randomly generated as a mixture of Gaussian distributions. In each class, data points are sampled *i.i.d.* from a multivariate Gaussian distribution with the covariance being σI, with I as the identity matrix. This setting indicates independent covariates of the data. We set the number of classes to be 4, the number of samples to be 400, and the number of dimensions to be 24. We set the value of σ in the range of $\{0.8,\ 0.85,\ 0.9\}$ respectively. Following the experimental setup in [174], we make three replicates for each feature in the data to form 24 feature groups (each group has three replicated features). We randomly pick 6 feature groups to generate the data such that we can evaluate the capability of GroupSAM in identifying truly useful feature groups. To make the classification task more challenging, we add random noise drawn from uniform distribution $\mathcal{U}(0, \theta)$ where θ is 0.8 times the maximum value in the data. In addition,

Table 5: Classification evaluation of GroupSAM on the synthetic data. The upper half use 24 features groups, while the lower half corresponds to 300 feature groups.

	SVM	GaussianSVM	L1SVM	SAM	GroupSpAM	GroupSAM
$\sigma = 0.8$	0.943±0.011	0.935±0.028	0.925±0.035	0.895±0.021	0.880±0.021	**0.953±0.018**
$\sigma = 0.85$	0.943±0.004	0.938±0.011	0.938±0.004	0.783±0.088	0.868±0.178	**0.945±0.000**
$\sigma = 0.9$	0.935±0.014	0.925± 0.007	0.938±0.011	0.853± 0.117	0.883±0.011	**0.945±0.007**
$\sigma = 0.8$	0.975±0.035	0.975±0.035	0.975±0.035	0.700±0.071	0.275±0.106	**1.000**
$\sigma = 0.85$	0.975±0.035	0.975±0.035	0.975±0.035	0.600±0.141	0.953±0.004	**1.000**
$\sigma = 0.9$	0.975±0.035	0.975±0.035	0.975±0.035	0.525±0.035	0.983±0.004	**1.000**

we test on a high-dimensional case by generating 300 feature groups (*e.g.,* a total of 900 features) with 40 samples in a similar approach.

We summarize the classification performance comparison on the synthetic data in Table 5. From the experimental results we notice that GroupSAM outperforms other approaches under all settings. This comparison verifies the validity of our method. We can see that GroupSAM significantly improves the performance of SAM, which shows that the incorporation of group information is indeed beneficial for classification. Moreover, we can notices the superiority of GroupSAM over GroupSpAM, which illustrates that our GroupSAM model is more suitable for classiciation. We also present the comparison of feature groups in Table 6. For illustration purpose, we use the case with 24 feature groups as an example. Table 6 shows that the feature groups identified by GroupSAM are exactly the same as the ground truth feature groups used for synthetic data generation. Such results further demonstrate the effectiveness of GroupSAM method, from which we know GroupSAM is able to select the truly informative feature groups thus improve the classification performance.

Table 6: Comparison between the true feature group ID (for data generation) and the selected feature group ID by GroupSAM on the synthetic data.

	True Feature Group IDs	Selected Feature Group IDs via GroupSAM
$\sigma = 0.8$	2,3,4,8,10,17	3,10,17,8,2,4
$\sigma = 0.85$	1,5,10,12,17,21	5,12,17,21,1,10
$\sigma = 0.9$	2,6,7,9,12,22	6,22,7,9,2,12

3.4.2 Performance Comparison on Benchmark Data

Here we use 7 benchmark data from UCI repository [82] to compare the classification performance. The 7 benchmark data includes: Ecoli, Indians Diabetes, Breast Cancer, Stock, Balance Scale, Contraceptive Method Choice (CMC) and Fertility. Similar to the settings in synthetic data, we construct feature groups by replicating each feature for 3 times. In each feature group, we add random noise drawn from uniform distribution $\mathcal{U}(0, \theta)$ where θ is 0.3 times the maximum value.

We display the comparison results in Table 7. We find that GroupSAM performs equal or better than the compared methods in all benchmark datasets. Compared with SVM and L1SVM, our method uses additive model to incorporate nonlinearity thus is more appropriate to find the complex decision boundary. Moreover, the comparison with Gaussian SVM and SAM illustrates that by involving the group information in classification, GroupSAM makes better use of the structure information among features such that the classification ability can be enhanced. Compared with GroupSpAM, our GroupSAM model is proposed in data dependent hypothesis spaces and employs hinge loss in the objective, thus is more suitable for classification.

3.4.3 MCI Conversion Prediction

In this chapter, we compare the methods on the problem of MCI conversion classification using the ADNI data as used in Chapter 2. We use the neuroimaging data collected as the baseline time to predict the progression status of the samples. All 396 samples with no missing MRI/PET features are included in this study, including 101 health control (HC) samples, 202 MCI samples and 93

Table 7: Classification evaluation of GroupSAM on benchmark data.

	SVM	GaussianSVM	L1SVM	SAM	GroupSpAM	GroupSAM
Ecoli	0.815±0.054	0.818±0.049	0.711±0.051	0.816±0.039	0.771±0.009	**0.839±0.028**
Indians Diabetes	0.651±0.000	0.652±0.002	0.638±0.018	0.652±0.000	0.643±0.004	**0.660±0.013**
Breast Cancer	**0.968±0.017**	0.965±0.017	0.833±0.008	0.833±0.224	0.958±0.027	0.966±0.014
Stock	0.913±0.001	0.911±0.002	0.873±0.001	0.617±0.005	0.875±0.005	**0.917±0.005**
Balance Scale	0.864± 0.003	0.869±0.004	0.870±0.003	0.763±0.194	0.848±0.003	**0.893±0.003**
CMC	0.420± 0.011	0.445±0.015	0.437±0.014	0.427±0.000	0.433±0.003	**0.456±0.003**
Fertility	**0.880± 0.000**	**0.880±0.000**	0.750±0.184	0.860±0.028	0.780±0.000	**0.880±0.000**

AD samples. In these 202 MCI samples, 126 of them remain MCI along the three-year continuum (*e.g.,* stable MCI), while the other 76 become AD in M36(*e.g.,* progressive MCI).

The results are summarized in Table 8. The goal of the experiment is to accurately classify subjects from four classes, including HC (health control), sMCI (stable MCI), pMCI (progressive MCI) and AD.From the comparison we notice that GroupSAM outperforms all other methods on both data, which confirms the effectiveness of our model. Compared with LinearSVM, GroupSAM is more appropriate for the complex non-linear classification in MCI conversion prediction. The comparison with GaussianSVM and PolynomialSVM indicates the flexibility of GroupSAM, since our model reduces variance of GaussianSVM as well as PolynomialSVM models. Moreover, since GroupSAM considers the interaction of imaging biomarkers and incorporates more information, thus proves to be more suitable than SAM in MCI conversion prediction.

3.4.4 Interpretation of Imaging Biomarker Interaction

Here we assess the important neuroimaging features and interactions learned by our model. The weights of interactions are derived directly from the α parameter, while the weights of features are calculated by summing up all interaction weights where the certain feature is involved.

Using the results via MRI data as an example, we plot the heat map of the interactions and features in Figure 7 and Figure 5. Also, we map the top 10 imaing biomarkers to the brain and

Table 8: Classification evaluation of GroupSAM on MRI and PET data for MCI conversion prediction.

Methods	**MRI**	**PET**
LinearSVM	0.424±0.010	0.386±0.006
RBF-SVM	0.429±0.011	0.429±0.003
PolynomialSVM-Quadratic	0.422±0.114	0.429±0.004
PolynomialSVM-Cubic	0.419±0.004	0.3866±0.071
SAM	0.456±0.222	0.753±0.020
GroupSAM	**0.551±0.004**	**0.760±0.066**

present the cortical map in Figure 6. In Figure 7 we notice that the interaction between lateral ventricle and occipital lobe may be important in AD. We infer the interaction between these two regions may be due to their positional relationship. In [5], significant enlargement of posterior lateral ventricle horns were reported in Parkinson's disease, thus it might be interesting to look into this phenomenon in AD patients.

Figure 5 indicates the importance of occipital lobe in MCI conversion. The influence of occipital lobe to AD has been studied in [15], which verified less WM than control in AD subjects. Such results suggest the validity of our model. More importantly, the findings of our model provide insights in better understanding the impact of different regions of interest (ROIs) in AD progression.

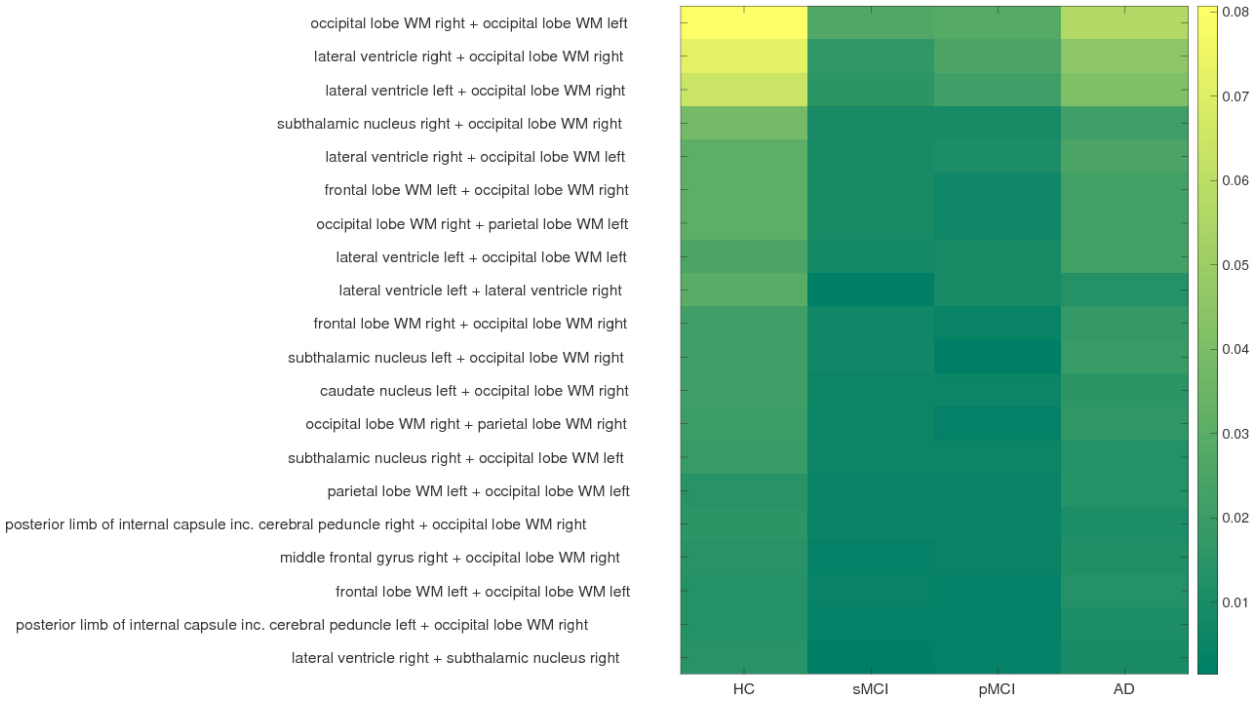

Figure 7: Heat maps of the top 20 MRI imaging marker interactions learned by GroupSAM.

4.0 Deep Neural Network for Large-Scale Data

4.1 Introduction

Gene expression profiling is a powerful tool for measuring the expression of thousands of genes under a given biological circumstance. It provides a comprehensive view of cellular status and is therefore the basis for functional gene expression pattern characterization. Gene expression profiling has been widely used in the analysis of various cell conditions and regulatory mechanisms in response to different disturbances, thereby enabling the discovery of cellular functionality and differentiation during the pathogenesis of disease.

The rapid development of high-throughput technologies has contributed to the proliferation of large-scale gene expression profiles. Several public databases have been constructed to archive gene expression data for various biological states. For example, Gene Expression Omnibus (GEO) [36] is a versatile data warehouse storing the gene expression measurement in different cells. The Connectivity Map (CMap) provides a collection of gene expression profiles from curated human cells. The availability of these databases help to improve the understanding of gene expression patterns in different cellular situations.

Gene expression analysis has facilitated recent biological studies in various fields, such as cancer classification and subtype discovery [14], estrogen-receptor (ER) status determination [98], drug-target network construction [164], cell type detection [30] as well as the influenza infection susceptibility and severity analysis [160]. By assessing global gene expression profiles in colorectal cancer (CRC) samples, researchers in [14] established the connection between elevated expression of mesenchymal genes in stromal cells and poor prognosis and resistance to therapy. On the basis of mRNA expression assay, Mouttet et al. developed a new quantitative assessment of hormone receptor status and HER2 status that characterizes various estrogen-dependent growth mechanisms in different menopausal states of ER-positive breast cancer [98]. By integrating gene expression microarray data, researchers in [164] constructed a bipartite graph to analyze the association between drug targets and disease-gene products, providing a clue to new drug discovery.

Moreover, microarray analysis in [160] identified significantly altered expression levels of several immune-related genes in mice susceptible to influenza A virus (IAV) infection. Despite the rapid advances and widespread application of gene expression profiling, genome-wide profiling remains expensive and difficult when it comes to analyzing numerous cell types in response to different interferences [100]. Therefore, how to accurately and efficiently evaluate the entire genome expression is still a key issue. According to previous studies, the expression patterns of different genes are highly correlated [124, 102, 53]. As is indicated in the cluster analysis of single-cell RNA-seq in [102] and [124], genes from the same cluster exhibited similar expression patterns under different conditions. Given such a high correlation among gene expression profiles, it is reasonable to assume that only a small group of genes can be informative to approximate the overall genome expression. To determine the appropriate small subset of informative genes, researchers in the Library of Integrated Network-based Cell-Signature (LINCS) plan (http://www.lincsproject.org/) performed principle component analysis (PCA) and identified $\sim$1000 genes that were sufficient to describe $\sim$ 80% of the information in the entire transcriptome [35]. This set of $\sim$1000 genes, called landmark genes, contains most of the information in the whole genome and can be used to predict the expression of other genes.

Based on the above findings, one credible and cost-effective strategy for large-scale gene expression profiling is to measure the expression profile of only landmark genes and then estimate the remaining target gene expression through an appropriate predictive model. Therefore, it is essential to construct effective computational methods to infer the target gene expression profiles from the landmark genes. The estimation of target gene expression profiles can be naturally formulated as a multi-task regression problem, where the prediction of one target gene can be formulated as one task. The most straightforward model is linear regression, which has been applied in the LINCS program. The LINCS program generated the landmark gene expression of $\sim$1.3 million profiles using L1000 technology, and adopt the linear regression model to infer the expression of the remaining target genes.

However, the regulatory network among genes is complicated, linear models do not have enough capacity to capture the non-linear relationship of the gene expression profiles [51]. Kernel models provide a way to introduce flexibility in representing the non-linear relations among gene expression. However, in large-scale scenarios, kernel methods need to calculate an extremely

large kernel matrix and therefore suffer from a high computational burden. In contrast, deep learning models are scalable and highly flexible, and have been widely applied to different biological problems, such as protein structure prediction [89], cancer classification [39], and population stratification detection [117]. The remarkable predictive power and flexibility of the deep learning model makes it a powerful alternative for effective inference large-scale gene expression profiles.

[20] applied deep neural networks to the multi-task regression problem for gene expression inference. The authors constructed a fully connected neural network (abbreviated as D-GEX) that outperformed linear models. The success of the D-GEX model proves the prospect of deep learning models in driving the gene expression inference problem. However, D-GEX model uses standard Mean Squared Error (MSE) loss function, which produces smooth and blurry results [92]. In other words, the use of MSE loss makes the model not capable of learning the high-frequency patterns in the data, thus performs poorly when data comes from multi-modal distribution. Also, training D-GEX model using the MSE loss is sensitive to outliers in the data, hence D-GEX is not a robust model. To deal with these problems, we propose a novel conditional generative model for robust and sharp estimation in the regression task. We consider adversarial loss found in Generative Adversarial Nets (GAN) [48] to estimate the target gene expression in a sharp and realistic approach. Moreover, we adopt ℓ_1-norm loss to stabilize the adversarial training and make our model robust to outliers. We apply our model for predicting target gene expression profiles from two different gene expression data portal: GEO and Genotype-Tissue Expression (GTEx) [87]. Our model significantly outperforms previous methods on the inference of gene expression, and also provides insights into the correlation between different genes.

Moreover, previous methods still suffers from several problems: 1) traditionally, gene expression inference is formulated as a regression problem, where the computational models attempt to approximate the conditional probability distribution of target genes given landmark genes, but do not consider their joint distribution; 2) previous methods formulate the gene expression inference in a totally supervised manner, where only profiles with both landmark and target gene expression measurements (named as "labeled" data according to the notations in previous chapter [155]) are involved in the training process. However, since the measurement of only landmark genes are much cheaper, there are a lot more profiles with the measurement of only landmark genes (named as "unlabeled" data according to the notations in [155]) are not used in the training process.

In order to solve these problems, we propose a novel semi-supervised generative adversarial network (abbreviated as SemiGAN) for gene expression inference. Our model is inspired by the inpainting problem in computer vision applications, where the goal is to fill in the missing part in a corrupted image based on the known image context and the learned distribution over the entire image. Here we regard the target gene expression as the missing part in a profile and the goal is to fill in the missing given the landmark gene information (*i.e.,* context). We propose to construct a deep generative model that approximate the joint distribution of landmark and target genes. By doing this, we analyze the overall distribution and correlation among genes which improves the inference. Moreover, we formulate our model in a semi-supervised manner that incorporates the profiles with only landmark genes into the training process. The use of the unlabeled section of data can improve the learning of landmark gene distribution and also strengthens the inference of target gene expression.

We would like to point out our main contributions as follows:

- Proposing a novel generative adversarial network for the problem of gene expression inference;
- Proposing a novel semi-supervised framework for gene expression inference;
- Introducing an effective loss function consisting of the adversarial and ℓ_1-norm losses for training the gene regression model;
- Introducing the collaborative training of our GAN and inference network;
- Outperforming alternative models with significant margins on two datasets according to different evaluation metrics

4.2 Related Work

4.2.1 Gene Expression Inference

Although rapid progress has been observed in high-throughput sequencing and analysis techniques, genome-wide expression profiling for large-scale libraries under different disturbance remains expensive and difficult [100]. Therefore, how to keep a low budget while the informative measurement in gene expression profiling remains a key issue. Previous studies have detected a

high degree of correlation among gene expression such that genes with similar function preserved similar expression patterns under different experimental circumstances. Due to the correlation structure existing in gene expression patterns, even a small number of genes can provide a wealth of information. Shah et al. [124] found that a random collection of 20 genes captured $\sim$50% of the relevant information throughout the genome. Recent advances in RNA-seq [102, 53] also support the notion that a small number of genes are abundant enough to approximately depict the overall information throughout the transcriptome.

Researchers from the LINCS program assembled GEO data on the basis of the microarray Affymetrix HGU133A to analyze the gene correlation structure and identify the subset of informative genes to approximate the overall information in genome. They collected the expression profiles from a total of 12,063 genes and determined the maximum percentage of correlation information can be recovered given a specific number of genes. The calculation of recovery percentage is based on the comparable rank from the Kolmogorov-Smirnov statistic. According to the LINCS analysis, researchers found that only 978 genes were capable of restoring 82% of the observed connections across the entire transcriptome [68]. The set of 978 genes have been characterized as landmark genes and can be used to deduce the expression of other target genes in different cell types under various chemical, genetic and disease conditions.

4.2.2 Deep Neural Networks

In recent years, deep learning has shown remarkable results in wide range of applications, such as computer vision [74], natural language processing [25], speech recognition [54], and even biological science[32]. The impressive capability of deep models is due to efficient and scalable leaning of discriminative features from raw data via multi-layer networks. Among different models, Goodfellow *et. al.* proposed a powerful generative model, called generative adversarial networks (GAN) [48], especially in computer vision tasks. In particular, GAN consists of two sub-networks, a generator and a discriminator, and aims to play minimax game between these networks. While the generator's goal is to fool the discriminator by synthesizing realistic images from arbitrary input (*i.e.* data from random noise distribution), the discriminator tries to distinguish between the real and synthesized (i.e. fake) images. In recent years, GAN model has been widely applied

to various tasks, including image generation [31, 67], image translation [175], semi-supervised image classification [119], image inpainting [107, 163], also speech enhancement [106] and drug discovery [12] and achieved good performance.

We also adopt GAN architecture in our model in order to learn the joint distribution of landmark and target genes. In one view, our model on inferring the target genes from landmark genes is similar to image inpainting methods [107, 163], in which the goal is to deduce the missing part in a corrupted image. Pathak et al. [107] employed the autoencoder architecture, where the encoder maps the corrupted image to a latent variable, and the decoder recovers the original image without damage. The framework attempted to reduce the reconstruction loss as well as the adversarial loss such that the recovered images followed similar distribution as real images. In another view, our work is similar to the semi-supervised image classification methods [119, 21], in which the task is to predict categorical labels of input image data. For instance in [21], GAN is utilized to learn the joint distribution of image and categorical labels in order to improve the classification task by the synthesized image-label pairs. However, our proposed model has major differences compared to the previous works. First, our task is semi-supervised regression on non-structured gene data, which is different from supervised inpainting and structured image data. Moreover, our generative model is unique in comparison with other models, since we train it using adversarial, reconstruction and translation loss functions.

In an unlabeled gene expression profile with the measurement of only landmark genes available, we treat the expression of target genes as the missing part in the profile. Our model uses the generator to recover the target gene expression profiles given the landmark gene expression such that the generated profile obeys similar distribution as real labeled profiles (with both landmark and target gene expression measured). However, we would like to emphasize two major differences in our model from [163]: firstly, our model is based on a semi-supervised learning framework, where we make use of the unlabeled profiles to strengthen the learning of data distribution; secondly, we consider both the conditional and joint distribution of landmark and target genes in our model, such that the learning process of these two distributions improve each other thus makes better prediction. Although our model falls into the category of conditional GAN models, it is different from the previous works due to the new application of gene expression inference and the challenges in generating large dimension outputs with no spatial structure.

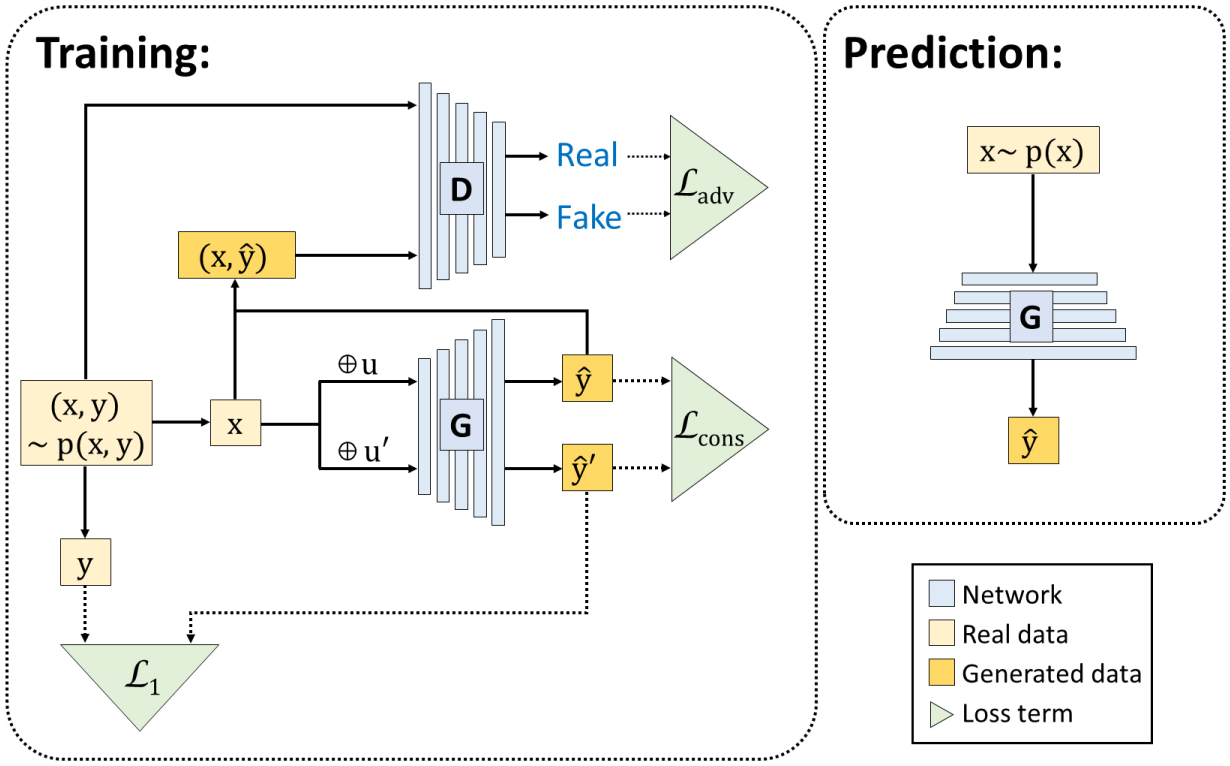

Figure 8: Illustration of GGAN architecture and its loss functions.

4.3 Conditional Generative Adversarial Network

4.3.1 Motivations

Given a set of gene expression profiles $\Omega = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, where $\{\mathbf{x}_i\}_{i=1}^n$ denotes n landmark gene expression profiles, and $\{\mathbf{y}_i\}_{i=1}^n$ corresponds to target genes, our goal is to learn a multi-task regression model for mapping landmark genes to the corresponding target genes $G : \mathbf{x} \rightarrow \mathbf{y}$, that is appropriate for the inference of each $\mathbf{y}_i$ given $\mathbf{x}_i$.

Although, the mean squared error (MSE) loss is the first objective candidate for learning this mapping function, it suffers from different problems. For instance, if the prediction probability of target genes for a landmark gene $\mathbf{x}$ has two equally likely modes $\mathbf{y}$ and $\mathbf{y}'$, then the average value

$\mathbf{y}_{ave} = (\mathbf{y} + \mathbf{y}')/2$ will be the estimation with the minimum MSE loss, even if the the $\mathbf{y}_{ave}$ itself has very low probability. In other words, MSE loss makes the estimation as the average over possible modes, thus leads to blurry and smooth prediction results.

To address the inherently smooth predictions obtained from the MSE loss function, we propose a novel deep generative model, denoted by GGAN, for the inference of target gene expression from landmark genes. In particular, we adopt a conditional adversarial network as our model, where the generator plays the role of conditional distribution of the target genes given the landmark genes, and the discriminator assesses the quality of generated target genes compared to the ground truths. Considering $\hat{\mathbf{y}}_i = G(\mathbf{x}_i)$ as the predicted target genes by the generator network, we train the discriminator to distinguish the real pairs $(\mathbf{x}, \mathbf{y})$ from the fake pairs $(\mathbf{x}, \hat{\mathbf{y}})$, and learn the generator to synthesize as realistic as possible $\hat{\mathbf{y}}$ samples to fool the discriminator.

In order to train the generator network, we combine the adversarial loss and ℓ_1-norm loss functions. In contrast to the smoothing effect of MSE loss, adversarial loss selects a single mode and results in sharp predictions. The ℓ_1-norm loss provides robust predictions to the outliers, and is also helpful in stabilizing the adversarial training. In another point of view, ℓ_1-norm loss function captures the low frequency structure of samples, and adversarial loss learns the high frequency parts of the data. Moreover, to guarantee that the output of our mapping function is stable *w.r.t.* the perturbation of random noises, we introduce consistency loss in our model such that the output should be similar when the input is added with different random noises. To make the motivation clear, we show the architecture of our model along with the applied loss functions in Figure 8.

4.3.2 Deep Generative Model

The min-max adversarial loss for training the generator and discriminator networks in our model has the following form.

$$\min_G \max_D \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}\big[\log(D(\mathbf{x},\mathbf{y}))\big] + \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\big[\log(1 - D(\mathbf{x}, G(\mathbf{x}))\big] \tag{4.1}$$

Note that the input to the discriminator network is the concatenation of landmark and target genes. This formation helps the generator to learn the joint distribution $p(\mathbf{x}, \mathbf{y})$, thus produces the corresponding target genes to the landmark genes. It also guides the discriminator to learn the relationship between the landmark and target genes.

The ℓ_1-norm loss function for training the generator network is:

$$\min_G \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}\big[\|\mathbf{y} - G(\mathbf{x})\|_1\big] \tag{4.2}$$

We also define the consistency loss for training the parameters of the generator as follows.

$$\min_G \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}\big[\|G(\mathbf{x}\oplus\mathbf{u}) - G(\mathbf{x}\oplus\mathbf{u}')\|^2\big], \tag{4.3}$$

where $\mathbf{u}$ and $\mathbf{u}'$ are the dropout noises that we add to the input and hidden layers of the generator network.

Training the GAN models to generate the large dimension samples using the adversarial loss is very challenging. There are some studies that propose tricks like patch-GANs [175], in which the discriminator only sees a small patch of input image, or progressive GAN [67], in which the generator network is expanded by adding layers during training and the size of generated images is increased as a result. However, we cannot use these tricks, since they are developed for the image data with spatial structure. In order to tackle this issue, we develop the idea of multiplying a binary mask to the inputs of discriminator network. The mask is constructed using the random Bernoulli distribution with probability p_{mask}, having 0 and 1 elements. We start training using the mask with the high probability (*i.e.* having more zero elements) to only show the small portion of genes to the discriminator, and then progressively decrease the probability to finally show all the genes to the discriminator at the end of training process. The empirical approximation of the adversarial loss with the incorporated mask has the following form:

$$\mathcal{L}_{adv} = \frac{1}{n}\sum_{i=1}^{n} \log(D(\mathbf{x}_i, \mathbf{y}_i\mathbf{m}_i)) + \log(1 - D(\mathbf{x}_i, G(\mathbf{x}_i)\mathbf{m}_i)), \tag{4.4}$$

where $\mathbf{m}_i$ represents the mask. Note that the mask only applies to the target genes in order to simplify the generation task. Also, the masks for $(\mathbf{x}_i, \mathbf{y}_i)$ and $(\mathbf{x}_i, G(\mathbf{x}_i))$ in each training step are same in order to show same set of target genes to the discriminator. Besides, we scale up the values of target genes by multiplying by $1/p_{mask}$ during training to keep the output activation intact. This mask not only increase the difficulty level of generation task progressively and stabilize the adversarial learning, but also considers the target genes conditionally independent.

Algorithm 2 Optimization of GGAN via mini-batch SGD method.

Input: Input gene expression profile $\Omega = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, where $\{\mathbf{x}_i\}_{i=1}^n$ denotes n landmark gene expression profiles, and $\{\mathbf{y}_i\}_{i=1}^n$ corresponds to target genes. Hyper-parameter λ_{adv} and λ_{cons}.

Output: Generator function G and discriminator function D.

1: **Initialize** parameters θ_D for D and parameters θ_G for G

2: **for** number of training iterations **do**

3: **for** t = 1,. . . , T **do**

4: randomly choose mini-batch $\Omega_t \subset \{1, \ldots, n\}$ of size b

5: Update D by ascending along its stochastic gradient:

$$\bigtriangledown_{\theta_D} \mathcal{L}_{adv}(D; \Omega_t).$$

6: Update G by descending along its stochastic gradient:

$$\bigtriangledown_{\theta_G} \mathcal{L}_{tot}(G; \Omega_t).$$

7: **end for**

8: **end for**

The ℓ_1-norm and consistency loss functions can be also approximated by the following empirical losses.

$$\mathcal{L}_1 = \frac{1}{n} \sum_{i=1}^n ||\mathbf{y}_i - G(\mathbf{x}_i)||_1 \tag{4.5}$$

$$\mathcal{L}_{cons} = \frac{1}{n} \sum_{i=1}^n \|G(\mathbf{x} \oplus \mathbf{u}) - G(\mathbf{x} \oplus \mathbf{u}')\|^2 \tag{4.6}$$

Combining the three loss terms in Eqs. (5.5), (4.5) and (4.6), we define the joint loss for training the generator network as

$$\mathcal{L}_{tot} = \mathcal{L}_1 + \lambda_{adv} \mathcal{L}_{adv} + \lambda_{cons} \mathcal{L}_{cons} \tag{4.7}$$

where λ_{adv} and λ_{cons} are the hyper-parameters to balance the role of different loss terms.

To update the parameters in generator G and discriminator D, we adopt gradient-based optimization, which is the most popular method for optimizing neural networks. The stochastic

gradient descent (SGD) methods are efficient in calculating the gradient yet introduce high variance in parameter updating thus leads to heavy fluctuation in the objective function. To handle this problem, mini-batch SGD methods propose to update the parameter θ *w.r.t.* each mini-batch $\Omega_m = \{(\mathbf{x}_{t_i}, \mathbf{y}_{t_i})\}_{i=1}^{n_t}$ given a general cost function $\mathcal{J}(\theta; \Omega)$ as follows:

$$\theta = \theta - \alpha \bigtriangledown_\theta \mathcal{J}(\theta; \Omega_t) \tag{4.8}$$

where $\Omega = \bigcup_{t=1}^{T} \Omega_t$ and any two mini-batches are disjoint.

In our gene expression inference problem, we adopt a variant of mini-batch SGD methods to update the parameters in generator G and discriminator D for an efficient and stable update. We summarize the optimization steps in Algorithm 2.

4.4 Generative Network for Semi-Supervised Learning

4.4.1 Problem Definition

In the gene expression inference problem, we use vector **x** to denote the landmark gene expression profile and vector **y** for the target gene expression. $\Omega_l = \{(\mathbf{x}_i^l, \mathbf{y}_i^l)\}_{i=1}^{n_l}$ collects the labeled profiles where the measurement for both landmark and target genes are available, while $\Omega_u = \{\mathbf{x}_j^u\}_{j=1}^{n_u}$ corresponds to the unlabeled profiles with the expression of only landmark genes measured. Usually we have $n_u \gg n_l$, since the measurement of only landmark genes is much cheaper than all the genes in the entire transcriptome. Our goal is to construct a model, which appropriately predicts the target gene expression using a small set of labeled genes (*i.e.* paired landmark and target genes) and a large set of unlabeled genes (*i.e.* landmark genes).

4.4.2 Motivation

In previous works, the inference of target gene expression is formulated as a multi-task regression, where predicting the expression of each target gene in **y** via landmark genes **x** is one regression task. The regression framework is usually formulated in a fully supervised manner, such that a

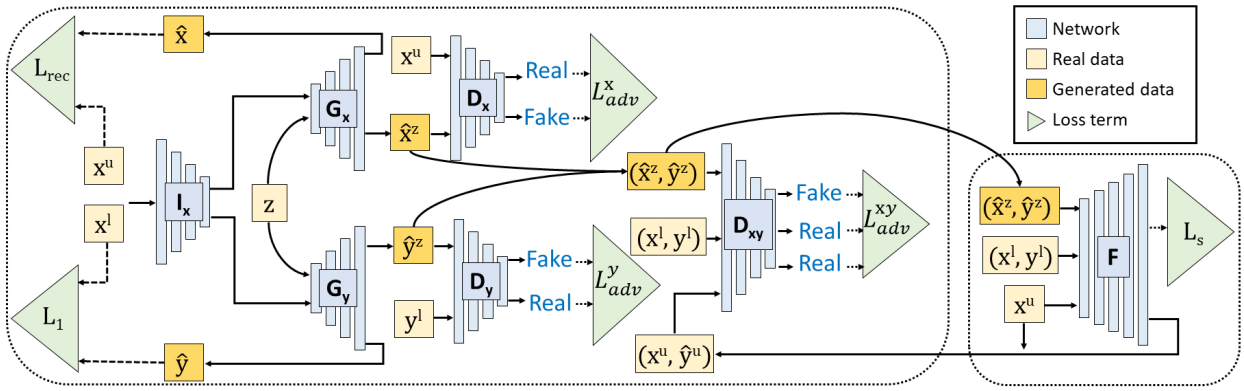

Figure 9: Illustration of the SemiGAN architecture for gene expression inference.

large set of labeled data is required to efficiently train the regression model. However in our problem, collecting the whole gene expression profiles (*i.e.* paired landmark and target genes $(\mathbf{x}, \mathbf{y})$) is much more expensive than the the landmark genes $\mathbf{x}$ alone. In order to address this issue and benefit from the plentiful unlabeled profiles, we propose a semi-supervised learning framework to take advantage of both labeled and unlabeled profiles and use the unlabeled data to strengthen the learning. Our proposed model consists of an inference network and a GAN sub-model. Generally, we consider the GAN sub-model to learn the joint distribution $p(\mathbf{x}, \mathbf{y})$, and the inference network to learn the conditional distribution $p(\mathbf{y}|\mathbf{x})$. We provide a collaboration framework between the GAN and inference networks, such that the GAN generates the approximated paired samples $(\hat{\mathbf{x}}^z, \hat{\mathbf{y}}^z)$ as reliable extra labeled data for training the inference network, and the approximated pairs $(\mathbf{x}^u, \hat{\mathbf{y}}^u)$ by the inference network improves the adversarial training of the GAN network.

In particular, our GAN network includes two generators G_x and G_y to synthesize both landmark genes $\hat{\mathbf{x}}^z$ and target genes $\hat{\mathbf{y}}^z$ from a shared random input $\mathbf{z}$ respectively, and three discriminators D_x, D_y, D_{xy} to distinguish between the real and fake data $\mathbf{x}^u$ vs. $\hat{\mathbf{x}}^z$, $\mathbf{y}^l$ vs. $\hat{\mathbf{y}}^z$, and $(\mathbf{x}^l, \mathbf{y}^l)$ vs. $(\hat{\mathbf{x}}^z, \hat{\mathbf{y}}^z)$ respectively. In addition to adversarial loss, we use a reconstruction and a translation loss functions to help training of our generators. To do so, we consider a network to learn the inverse mapping of G_x, where the input is the landmark genes and the output has the same dimen-

sion of $\mathbf{z}$. Using this inverse network I_x, we define a reconstruction loss function for unlabeled data $\mathbf{x}^u$ through $I_x \rightarrow G_x$ pathway, and a translation loss function for labeled data $(\mathbf{x}^l, \mathbf{y}^l)$ through $I_x \rightarrow G_y$ pathway. Note that theses two loss functions are helpful in adversarial training of our generator networks, and aid generating large-dimension and unstructured gene data by avoiding mode collapse issue and using side information. Furthermore, we employ the inference network F to map the landmark gene expressions to the target gene expressions. For clarification purpose, we plot the architecture of our model, called SemiGAN, along with the applied loss functions in Figure 9.

4.4.3 Semi-Supervised GAN Model

As mentioned, SemiGAN has two generators and three discriminator networks. Following we show the adversarial loss functions corresponding to the pairs of generator and discriminator networks. The min-max adversarial loss for training the generator network G_x and discriminator network D_x is formulated as:

$$\min_{G_x} \max_{D_x} \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\big[\log(D_x(\mathbf{x}))\big] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[\log(1 - D_x(G_x(\mathbf{z}))\big] \tag{4.9}$$

where the goal is to learn the distribution of $p(\mathbf{x})$ via G_x, and generate realistic fake landmark gene samples.

The adversarial loss for training the generator network G_y and discriminator network D_y is formulated as:

$$\min_{G_y} \max_{D_y} \mathbb{E}_{\mathbf{y}\sim p(\mathbf{y})}\big[\log(D_y(\mathbf{y}))\big] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[\log(1 - D_y(G_y(\mathbf{z}))\big] \tag{4.10}$$

where the goal is to learn the distribution of $p(\mathbf{y})$ using G_y, and generate realistic fake target gene samples.

The min-max adversarial loss for training the networks D_{xy}, G_x, G_y is formulated as:

$$\begin{aligned}\min_{G_x,G_y} \max_{D_{xy}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}\big[\log(D_{xy}(\mathbf{x},\mathbf{y}))\big] \\ + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[\log(1 - D_{xy}(G_x(\mathbf{z}), G_y(\mathbf{z}))\big]\end{aligned} \tag{4.11}$$

The goal is to learn the corresponding relationship between the paired landmark and target gene expressions. Note that we consider the shared random input $\mathbf{z}$ for both generators to learn the joint distribution of landmark and target genes as $p(\mathbf{x}, \mathbf{y}|\mathbf{z})$. In addition to the labeled data $(\mathbf{x}^l, \mathbf{y}^l)$, we suppose $(\mathbf{x}^u, F(\mathbf{x}^u))$ as the real paired data in the above loss function, when the predictions of inference network are good enough after a few training epochs.

The auxiliary reconstruction loss function for training the inverse network I_x and the generator network G_x is:

$$\min_{I_x, G_x} \mathbb{E}_{(\mathbf{x})\sim p(\mathbf{x})} \left[\|\mathbf{x} - G_x(I_x(\mathbf{x}))\|_1 \right] \tag{4.12}$$

The auxiliary translation loss function for training I_x and G_y is:

$$\min_{I_x, G_y} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})} \left[\|\mathbf{y} - G_y(I_x(\mathbf{x}))\|_1 \right] \tag{4.13}$$

We also help training of the inverse network with the following loss:

$$\min_{I_x} \mathbb{E}_{(\mathbf{z})\sim p(\mathbf{z})} \left[\|I_x(G_x(\mathbf{z})) - \mathbf{z})\|_1 \right] \tag{4.14}$$

The loss function for training the inference network F is:

$$\begin{aligned} \min_{F} & \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})} \left[\|\mathbf{y} - F(\mathbf{x})\|_1 \right] + \mathbb{E}_{(\mathbf{z})\sim p(\mathbf{z})} \left[\|G_y(\mathbf{z}) - F(G_x(\mathbf{z}))\|_1 \right] \\ & + \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \left[\|F(\mathbf{x} \oplus \mathbf{e}) - F(\mathbf{x} \oplus \mathbf{e}')\|^2 \right] \end{aligned} \tag{4.15}$$

where the first term is the ℓ_1 loss using the original labeled data $(\mathbf{x}^l, \mathbf{y}^l)$, the second term is the ℓ_1 loss using the pseudo-labeled data $(\hat{\mathbf{x}}^z, \hat{\mathbf{y}}^z)$ synthesized by the generators, and the last term is the consistency loss that requires similar outputs for an input with different added noises $\mathbf{e}$ and $\mathbf{e}'$.

In our gene expression completion problem, we adopt a variant of mini-batch SGD methods to update the parameters in the networks for an efficient and stable update. We summarize the optimization steps of SemiGAN in Algorithm 3 by considering the empirical approximation of the expectations in the aforementioned loss functions.

4.5 Experimental Results

In this section, we apply our model to gene expression data from three different projects, *i.e.* Gene Expression Omnibus (GEO), Genotype-Tissue Expression (GTEx) and 1000G (1000 Genomes). The goal is to correctly predict the expression value of target genes based on the expression of landmark genes. In the meantime, we propose to interpret the role of each landmark gene in the inference of target gene expression, which may provide insights into the the information captured by the landmark genes as well as the correlation between different genes.

4.5.1 Experimental Setup

4.5.1.1 Datasets We download three different publicly available datasets from `https://cbcl.ics.uci.edu/public_data/D-GEX/` for this analysis, which includes: the GEO dataset based on microarray data, the GTEx dataset based on RNA-Seq data and the 1000 Genomes (1000G) RNA-Seq expression data.

The original GEO dataset consists of 129158 gene expression profiles corresponding to 22268 probes (978 landmark genes and 21290 target genes) that are collected from the Affymetrix microarray platform. The original GTEx dataset is composed of 2921 profiles from the Illumina RNA-Seq platform in the format of Reads Per Kilobase per Million (RPKM). While the original 1000G dataset includes 2921 profiles from the Illumina RNA-Seq platform in the format of RPKM.

We follow the pre-processing protocol in [20] for duplicate samples removal, joint quantile normalization and cross-platform data matching. Among the 22268 genes in the GEO data, there are 10463 genes having corresponding Gencode annotations in RNASeq. In the joint quantile normalization, we map the expression values in the GTEx and 1000G datasets according to the quantile computed in the GEO data. The expression value has been quantile normalized to the range between 4.11 and 14.97. Finally, the expression value of each gene has been normalized to zero mean and unit variance. After pre-processing, there are a total of 111009 profiles in the GEO dataset, 2921 profiles in the GTEx dataset while 462 profiles in the 1000G dataset. All the profiles correspond to 10463 genes (943 landmark genes and 9520 target genes).

4.5.1.2 Evaluation Criterion In the experiments, we use two different evaluation metrics, including mean absolute error (MAE) and concordance correlation (CC). Given a set of testing data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, for a certain model we denote the predicted expression set as $\{\hat{\mathbf{y}}_i\}_{i=1}^n$. The definition of MAE is:

$$MAE_t = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_{jt} - y_{jt}|, \tag{4.16}$$

where y_{it} represents the expression value for the t-th target gene in the i-th testing profile, and $\hat{y}_{it}$ indicates the corresponding predicted value. MAE_t is the MAE value for the t-th target gene.

The following equation shows the definition of CC:

$$CC_t = \frac{2\rho\sigma_{\mathbf{y}_t}\sigma_{\hat{\mathbf{y}}_t}}{\sigma_{\mathbf{y}_t}^2 + \sigma_{\hat{\mathbf{y}}_t}^2 + (\mu_{\mathbf{y}_t} - \mu_{\hat{\mathbf{y}}_t})^2}, \tag{4.17}$$

where CC_t indicates the concordance correlation for the t-th target gene. ρ is the Pearson correlation, while $\mu_{\mathbf{y}_t}$, $\mu_{\hat{\mathbf{y}}_t}$, and $\sigma_{\mathbf{y}_t}$, $\sigma_{\hat{\mathbf{y}}_t}$ are the mean and standard deviation of $\mathbf{y}_t$ and $\hat{\mathbf{y}}_t$ respectively.

4.5.1.3 Baseline Methods In the LINCS program, the gene expression inference is based on the least square regression (LSR) model:

$$\min_{W,\mathbf{b}} \sum_{i=1}^{n_l} ||W^T\mathbf{x}_i^l + \mathbf{b} - \mathbf{y}_i^l||^2 \tag{4.18}$$

where W is the weight matrix and $\mathbf{b}$ is the bias term. The learning is based on the labeled profiles $\Omega_l = \{(\mathbf{x}_i^l, \mathbf{y}_i^l)\}_{i=1}^{n_l}$. The LSR model is prone to overfit the training model, and therefore has limited prediction power. To deal with the overfitting problem, we also consider two other linear regression models in the comparison, which are ridge regression, *i.e.,* LSR with ℓ_2-norm regularization (LSR-L2) and LASSO regression, *i.e.,* LSR with ℓ_1-norm regularization (LSR-L1).

Besides the linear regression models, we also compare with the k nearest neighbor (KNN) method for regression, where the prediction of a given profile is formulated as the average of its k nearest profiles. Moreover, we compare with a deep learning method for gene expression inference (D-GEX) [20] to validate the performance of our SemiGAN model. The D-GEX model use a fully connected multi-layer perceptron for regression. To the best of our knowledge, D-GEX is the only model that apply deep learning frameworks to the gene expression inference problem.

Following the experimental settings in [20], we evaluate the methods under two different circumstances. Firstly, we use 80% of the GEO data for training, 10% of the GEO data for validation while the other 10% of the GEO data for testing. Secondly, we use the same 80% of the GEO data for training, the 1000G data for validation while the GTEx data for testing. Among the training data, we set the portion of labeled profiles to be $\{1\%, 3\%, 5\%, 10\%, 20\%\}$ respectively and leave the remaining as unlabeled. In the second scenario, the training, validation and testing comes from different platforms, which is designed to validate if comparing methods are capable of capturing the information for cross-platform prediction. We use the training data to construct the predictive model, validation data for model selection and parameter setting, while the testing data to conduct the evaluation. For LSR-L1 and LSR-L2 model, we tune the hyperparameter λ in the range of $\{10^{-2}, 10^{-1}, \ldots, 10^{3}\}$ according to the performance on the validation data. For each method, we report the average performance and standard deviation over all target genes on the testing data.

4.5.1.4 Implementation Details of GGAN We use similar architecture for the both datasets, train the networks only using the training sets, tune the hyper-parameters via the validation sets, and report the results on the test sets. For the generator network, we employ a DenseNet [57] architecture with three hidden layers, each one containing $9,000$ hidden units. For the discriminator, we use a fully connected network with one hidden layer including $3,000$ hidden units. We consider leaky rectified linear unit (LReLU) [90] with leakiness ratio 0.2 as the activation function of all layers except the last layer of generator network, which has linear function due to the mean-zero and unit-variance data normalization. Moreover, we set the maximum and minimum learning rates to 5×10^{-4} and 1×10^{-5} respectively, and linearly decrease it during training with the maximum epoch 500. Adam algorithm [71] is adopted as our optimization method with the default hyper-parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$. The batch size is set to 200. We also utilize weight normalization [120] as layer normalization to speed up the convergence of training process. The parameters of all layers are all initialized by Xavier approach [46]. We also select dropout, λ_{cons}, and λ_{adv} from $dropout^{set} = \{0.05, 0.1, 0.25\}$, $\lambda_{cons}^{set} = \{1, 10, 50\}$, and $\lambda_{adv}^{set} = \{0.1, 1, 5\}$ respectively. We use Theano toolbox for writing our code, and run the algorithm in a machine with one Titan X pascal GPU.

4.5.1.5 Implementation Details of SemiGAN We use networks with similar architecture for both of the two datasets, train the networks only using the training data, tune the hyper-parameters via the validation samples, and report the results on the test sets. For the inference network, we utilize a DenseNet [57] architecture with three hidden layers, each one containing $3,000$ hidden units. For the generators and discriminators, we use fully connected networks with three and one hidden layers respectively, where all the hidden layers include $3,000$ hidden units. The similar architecture to the generator is considered for the inverse network. We consider leaky rectified linear unit (LReLU) [90] with leakiness ratio 0.2 as the activation function of all layers except the last layer of generator network, which has linear function due to the mean-zero and unit-variance data normalization. Moreover, we set the maximum and minimum learning rates to 5×10^{-4} and 1×10^{-5} respectively, and linearly decrease it during training till the maximum epoch 500. Adam algorithm [71] is adopted as our optimization method with the default hyper-parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-08$. The batch size is set to 200. We also utilize weight normalization [120] as layer normalization to speed up the convergence of training process. The parameters of all layers are all initialized by Xavier approach [46]. We use Theano toolbox for writing our code, and run the algorithm in a machine with one Titan X pascal GPU.

Furthermore, we replace the original adversarial loss in the GAN models with the least-squares loss in [91]. We find that this loss leads to more stable training of our model, and paralytically provides better experimental results. In particular, we optimize the generator and discriminator parameters with the following least square loss instead of the sigmoid cross entropy loss function in Eq. (5.5).

$$\mathcal{L}_{adv}(D;\Omega) = \tfrac{1}{2}\sum_{i=1}^{n}[D(\mathbf{x}_i, \mathbf{y}_i\mathbf{m}_i) - 1]^2 + [D(\mathbf{x}_i, G(\mathbf{x}_i)\mathbf{m}_i)]^2 \tag{4.19}$$

$$\mathcal{L}_{adv}(G;\Omega) = \sum_{i=1}^{n}[D(\mathbf{x}_i, G(\mathbf{x}_i)\mathbf{m}_i) - 1]^2 \tag{4.20}$$

4.5.2 Prediction of GEO Data via GGAN

We first present the comparison results on the GEO data in Table 9. The results for GGAN model are obtained from the DenseNet architecture. We can observe apparent improvement of our model over other methods. First of all, deep learning models (D-GEX and GGAN) always performs better than linear models (LSR, LSR-L1 and LSR-L2), which indicates the superiority of

Table 9: Performance evaluation of GGAN on GEO data. The results of the comparing models are obtained by us running the released codes, except the one marked by ($\star$) on top that is reported from the original paper.

Methods	**MAE**	**CC**
LSR	0.376±0.084	0.823±0.096
LSR-L1	0.376±0.084	0.822±0.096
LSR-L2	0.376±0.084	0.822±0.096
KNN-GE	0.587±0.070	0.572±0.342
D-GEX	0.320±0.088* / 0.320±0.088	- / 0.869±0.090
GGAN	**0.290 ± 0.089**	**0.879±0.089**

Table 10: MAE comparison between D-GEX and GGAN model *w.r.t.* hidden layer and hidden units numbers for GEO data.

Methods	# hidden unit 3000	6000	9000	# hidden layers
D-GEX	0.342±0.086	0.334±0.087	0.330±0.087	1
	0.338±0.085	0.328±0.087	0.322±0.088	2
	0.336±0.085	0.325±0.087	0.320±0.088	3
GGAN	0.327±0.085	0.317±0.087	0.309±0.086	1
	0.316±0.085	0.304±0.087	0.298±0.087	2
	0.313±0.085	0.302±0.087	**0.297±0.087**	3

deep models in interpreting the non-linear association between different genes. Moreover, we can notice that GGAN model gains significantly better performance than D-GEX, which validates the success of applying the adversarial mechanism in the gene expression inference problem. Compared with D-GEX with mean squared error loss, our model considers both adversarial loss and ℓ_1-norm loss, thus make more sharp and realistic prediction results. Moreover, in order to show that

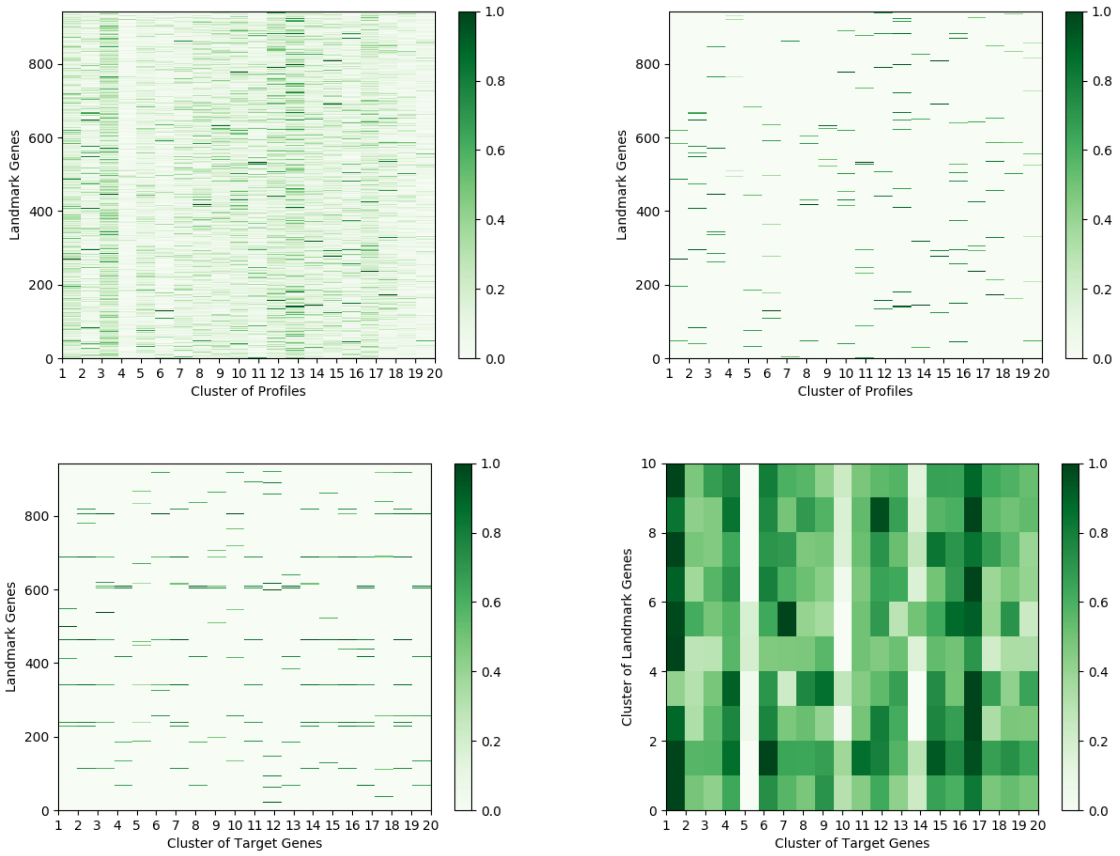

Figure 10: Heatmaps of the importance of landmark genes in the fully connected network of GGAN model on GEO data.

the major superiority of GGAN over D-GEX comes from the adversarial mechanism in our model design, we further compare D-GEX and GGAN with exactly the same structure (both models use fully connected network with varying number of hidden units and hidden layers). We present the comparison results in Table 10 and we can find GGAN consistently outperforms D-GEX regardless the setting of hidden layers and hidden units. This observation further validates that the use of adversarial loss and ℓ_1-norm loss in GGAN model overcomes the blurry and smooth prediction from D-GEX and make better inference for target genes.

Table 11: Performance evaluation of GGAN on GTEx data. The results of the comparing models are obtained by us running the released codes, except the one marked by ($\star$) on top that is reported from the original paper.

Methods	**MAE**	**CC**
LSR	0.470±0.124	0.718±0.207
LSR-L1	0.567±0.127	0.681±0.219
LSR-L2	0.468±0.123	0.718±0.208
KNN-GE	0.652±0.098	0.394±0.412
D-GEX	0.439±0.124* / 0.438±0.124	- / 0.734±0.207
GGAN	**0.422±0.126**	**0.748±0.207**

Table 12: MAE comparison between D-GEX and GGAN model *w.r.t.* hidden layer and hidden units numbers for GTEx data.

Methods	# hidden unit 3000	6000	9000	# hidden layers
D-GEX	0.451±0.123	0.443±0.125	0.439±0.125	1
	0.459±0.119	0.445±0.123	0.439±0.124	2
	0.516±0.116	0.460±0.119	0.449±0.121	3
GGAN	0.437±0.124	0.435±0.124	0.431±0.125	1
	0.441±0.124	0.432±0.124	0.430±0.124	2
	0.431±0.124	0.429±0.123	**0.429±0.124**	3

4.5.3 Prediction of GTEx Data via GGAN

Furthermore, in this subsection we present the results for the cross-platform prediction, where we use GEO data for training, 1000G data for validation while GTEx data for testing. We summarize the comparison results in Tables 11 and 12. Our model still gains significant advantage

over the comparing methods, which indicates that our GGAN model is capable of capturing the cross-platform information, such that the model constructed on the GEO data predict well for the inference of GTEx data.

4.5.4 Visualization of GGAN Network Relevance

In this subsection, we plot several visualization figures to show the role of different landmark genes in the gene expression inference problem. We adopt the Layer-wise Relevance Propagation (LRP) [8] method to calculate the importance of landmark genes. In Figure 10, we look into the results from the fully connected networks (structure for results in Table 10 on the GEO data. Firstly, we divide the gene expression profiles into 20 clusters and then use LRP to calculate the relevance score of landmark genes *w.r.t.* each profile cluster. Figure 10 (a) and (b) indicate that the landmark gene expression patterns for various profile groups are different, which replicates the findings in previous cancer subtype discovery and cancer landscape study that different group of samples usually exhibit different expression patterns [134, 65]. Next, we analyze the relationship between landmark genes and target genes. We cluster the target genes into 20 groups and calculate the overall relevance score of landmark genes in the prediction of each target gene cluster. For a clear visualization, we group the landmark genes into 10 clusters and display the association between landmark gene clusters and target gene clusters in Figure 10 (d). We can notice apparent difference in the relevance patterns for different target gene clusters, yet some similarity among certain clusters, *e.g.,* cluster (column) 5, 10 and 14. Cluster 5, 10 and 14 show consistent lower relevance value with the landmark genes, while cluster 14 shows higher correlation with the six-th landmark gene cluster than others. This finding has also been validated by previous gene cluster analysis [93], where gene cluster information is related to the structure of biosynthetic pathways and metabolites.

Moreover, we plot the illustration results on the prediction of GTEx data in Figure 11 and find similar result as in GEO data. It is notable that our model for the prediction of GTEx data is training on the GEO data, which validates that our model is able to appropriately capture the relation among genes for the cross-platform prediction.

4.5.5 Comparison on the GEO Data for SemiGAN

In this subsection, we evaluate the methods on the prediction of target gene expression in GEO data. From the summarization in Table 13 and 14, we can notice apparent improvement of our model over the counterparts. Firstly, we can find deep learning models (D-GEX and SemiGAN) consistently outperform all linear models (LSR, LSR-L1 and LSR-L2), since the deep neural network is capable of interpreting the non-linear association among gene expression patterns. Deep learning models indicate remarkable representation power to estimate the latent data distribution thus make better prediction for the expression of target genes. Besides, KNN shows worse results than the comparing methods, which is because of the inconsistence between the nearest neighbors in the training and testing data. Moreover, our SemiGAN model presents consistent advantage over the comparing deep model, D-GEX, due to the following two reasons: 1) the semi-supervised framework in our model enables the integration of unlabeled profiles in the learning, which strengthens the estimation of the data distribution and also introduces more data to train the inference network; 2) the estimation of both conditional distribution $p(\mathbf{y}|\mathbf{x})$ and joint distribution $p(\mathbf{x}, \mathbf{y})$ provides guidance for each other, such that the training of both generator and inference framework can be improved.

Moreover, we can notice that the superiority of SemiGAN model is more obvious with the labeled portion being 10% and 20%. When the labeled portion is too small, all methods are influenced by the limited number of labeled profiles. However, with just 10% labeled profiles available, the generators in our model can approximately estimate the joint distribution $p(\mathbf{x}, \mathbf{y})$ and produce reliable profiles to improve the learning of inference network. Conversely, the construction of the inference network also guide the generators to produce realistic gene expression profiles. This result validates that the SemiGAN can make good prediction of the target gene expression given very limited number of labeled profiles, which provides an accurate and cost-effective strategy for reliable genome-wide expression profiling.

4.5.6 Comparison on the GTEx Data for SemiGAN

Furthermore, we evaluate the comparing methods on the cross-platform prediction, where we use GEO data for training, 1000G data for validation while GTEx data for testing. This cross-

platform setting is used to test if the methods can capture appropriate information for predicting target gene expression from a different platform. As we can notice from the comparison results in Table 15 and 16, our SemiGAN model maintains significant advantage over the counterparts. Since the training and testing data come from different platform (*i.e.,* different data distribution), the performance on the GTEx data is not as good as the one for GEO data prediction. In the cross-platform prediction, SemiGAN still performs better, which validates that our model can take advantage of the learning of both conditional distribution and joint distribution to strengthen the cross-platform learning.

4.5.7 Analysis of Landmark Genes in SemiGAN Prediction

Here we look into the roles of landmark genes in the prediction of target gene expression. We use the Layer-wise Relevance Propagation (LRP) [8] method to calculate the importance of each landmark gene and plot the illustration figure in Figure 12. The LRP method calculates the relevance score for each landmark gene, where higher relevance score shows more contribution in the overall prediction of the target gene expression. Firstly, we analyze the contribution of each landmark gene for different profiles. Since there are a large number of profiles, we divide the them into 20 groups and show the accumulated relevance score pattern for each profile group in Figure 12 (a) and (b). We can notice that the landmark gene expression patterns vary for different profile groups, which replicates the previous findings in cancer clustering analysis that different group of cancer samples usually exhibit different expression patterns. The breast cancer subtype discovery study indicates different expression-based prognostic signatures for different subtypes [42]. And cancer landscape study also identified that cross-tissue cancer clusters can be characterized by different gene expression patterns [134].

Afterwards, we analyze the relationship between landmark genes and target genes. We cluster the target genes into 20 groups and calculate the relevance score for each target gene cluster, where we plot the overall contribution of each landmark gene across all profiles. To make a clear illustration, we group the landmark genes into 10 clusters and display the association between landmark gene clusters and target gene clusters in Figure 12 (d). Similar to the results between profiles and landmark genes, apparent difference in the relevance patterns can also be observed

for different target gene clusters. This finding has also been validated by previous gene cluster analysis [93], where gene cluster information is related to the structure of biosynthetic pathways and metabolites.

Algorithm 3 Optimization of SemiGAN via mini-batch SGD method.

Input: Labeled gene expression dataset $\Omega_l = \{(\mathbf{x}_i^l, \mathbf{y}_i^l)\}_{i=1}^{n_l}$ which corresponds to the profiles with the measurement of both landmark and target genes; and $\Omega_u = \{\mathbf{x}_j^u\}_{j=1}^{n_u}$ representing the profiles with only landmark gene expression measurement available. Hyper-parameter λ_{D_x}, λ_{D_y}, $\lambda_{D_{xy}}$, λ_{G_x}, λ_{G_y}, $\lambda_{G_{xy}}$, λ_{tra}, λ_{rec}, λ_{inv}, λ_{syn} and λ_{con}.

Initialize parameter θ_{D_x}, θ_{D_y} and $\theta_{D_{xy}}$ for discriminators, parameter θ_{G_x}, θ_{G_y} for generators, parameter θ_{I_x} for the inverse network I_x and parameter θ_F for the inference network F.

for number of training iterations **do**

  for t = 1,. . . , T **do**

    Randomly choose mini-batch $\Omega_l^t \subset \{1, \ldots, n_l\}$ of size b and mini-batch $\Omega_u^t \subset \{1, \ldots, n_i\}$ of size b.

    Update the parameters θ_{D_x}, θ_{D_y} and $\theta_{D_{xy}}$ by ascending along the stochastic gradient *w.r.t.* the following adversarial loss.

$$\max_{D_x, D_y, D_{xy}} \frac{1}{b} \sum_{i=1}^{b} \lambda_{D_x} \log(D_x(\mathbf{x}_i^u)) + \lambda_{G_x} \log(1 - D_x(G_x(\mathbf{z}_i))) + \lambda_{D_y} \log(D_y(\mathbf{y}_i^l))$$
$$+\lambda_{G_y} \log(1 - D_y(G_y(\mathbf{z}_i))) + \lambda_{D_{xy}} \log(D_{xy}(\mathbf{x}_i^l, \mathbf{y}_i^l))$$
$$+\lambda_{G_{xy}} \log(1 - D_{xy}(G_x(\mathbf{z}_i), G_y(\mathbf{z}_i)))$$

    Update the parameters θ_{G_x} and θ_{G_y} by descending along the stochastic gradient *w.r.t.* the following loss.

$$\min_{G_x, G_y} \frac{1}{b} \sum_{i=1}^{b} \lambda_{G_x} \log(1 - D_x(G_x(\mathbf{z}_i))) + \lambda_{G_y} \log(1 - D_y(G_y(\mathbf{z}_i)))$$
$$+ \lambda_{G_{xy}} \log(1 - D_{xy}(G_x(\mathbf{z}_i), G_y(\mathbf{z}_i))) + \lambda_{rec} ||G_x(I_x(\mathbf{x}_i^u)) - \mathbf{x}_i^u||_1$$
$$+ \lambda_{tra} ||\mathbf{y}_i^l - G_y(I_x(\mathbf{x}_i^l))||_1$$

    Update the parameters θ_{I_x} by stochastic gradient descent *w.r.t.* the following loss.

$$\min_{I_x} \quad \frac{1}{b} \sum_{i=1}^{b} \lambda_{rec} ||G_x(I_x(\mathbf{x}_i^u)) - \mathbf{x}_i^u||_1 + \lambda_{tra} ||\mathbf{y}_i^l - G_y(I_x(\mathbf{x}_i^l))||_1$$
$$+ \lambda_{inv} \|I_x(G_x(\mathbf{z})) - \mathbf{z})\|_1$$

    Update the parameters θ_F by stochastic gradient descent *w.r.t.* the following loss.

$$\min_{F} \quad \frac{1}{b} \sum_{i=1}^{b} ||G_y(\mathbf{z}_i) - F(G_x(\mathbf{z}_i))||_1 + \lambda_{syn} ||\mathbf{y}_i^l - F(\mathbf{x}_i^l)||_1$$
$$+ \lambda_{con} \|F(\mathbf{x}_i^u \oplus \mathbf{e}) - F(\mathbf{x}_i^u \oplus \mathbf{e}')\|^2$$

  end for

end for

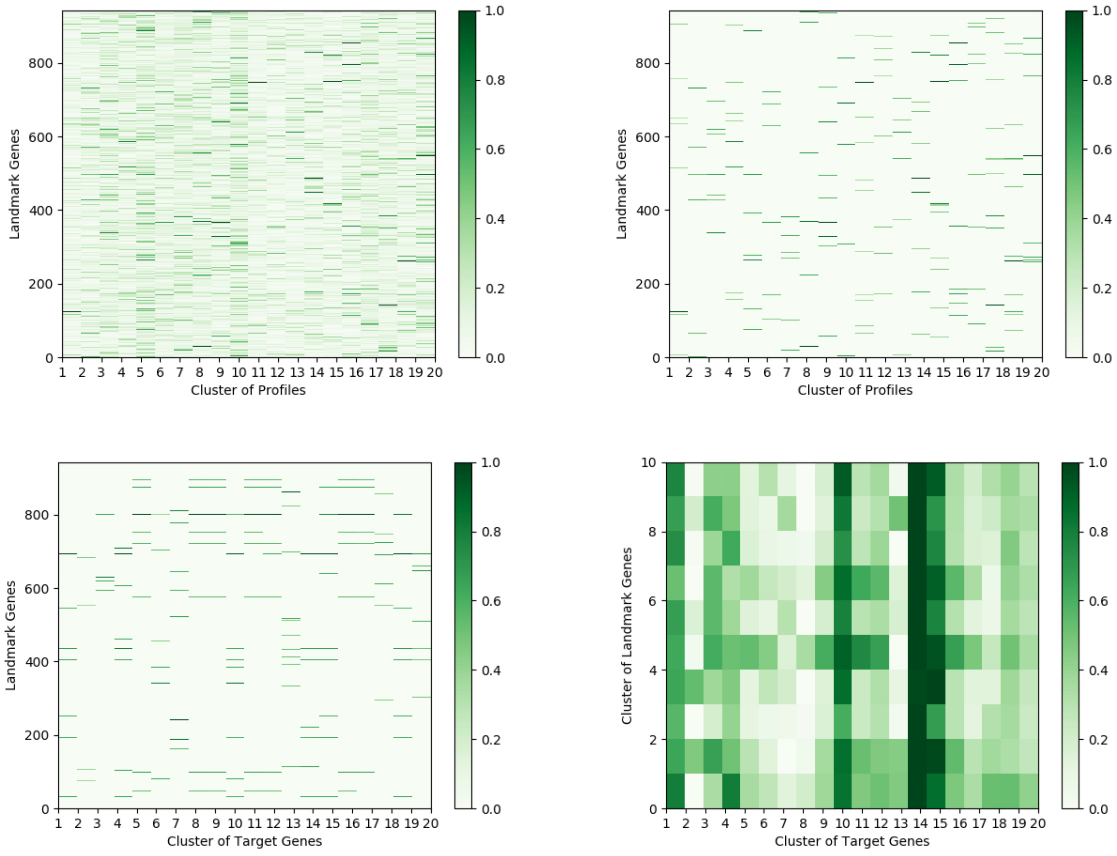

Figure 11: Illustration of the relevance score of different landmark genes calculated by the DenseNet architecture in GGAN for GTEx data.

Table 13: MAE comparison on GEO data with different portion of labeled data.

Methods	**1%**	**3%**	**5%**	**10%**	**20%**	**100%**
LSR	1.679±0.475	0.494±0.110	0.444±0.098	0.408±0.091	0.391±0.087	0.376±0.084
LSR-L1	0.451±0.092	0.418±0.084	0.412±0.082	0.407±0.081	0.405±0.081	0.376±0.084
LSR-L2	0.436±0.081	0.407±0.084	0.399±0.085	0.391±0.086	0.385±0.085	0.376±0.084
KNN	0.530±0.089	0.485±0.090	0.466±0.090	0.441±0.091	0.417±0.092	0.371±0.096
D-GEX	0.454±0.092	0.408±0.082	0.389±0.086	0.374±0.086	0.351±0.086	0.320±0.088
SemiGAN	**0.420±0.088**	**0.382±0.088**	**0.365±0.088**	**0.343±0.087**	**0.325±0.087**	**0.300±0.087**

Table 14: CC comparison on GEO data with different portion of labeled data.

Methods	**1%**	**3%**	**5%**	**10%**	**20%**	**100%**
LSR	0.243±0.121	0.741±0.121	0.777±0.111	0.801±0.104	0.812±0.100	0.823±0.096
LSR-L1	0.746±0.121	0.774±0.110	0.778±0.109	0.781±0.108	0.782±0.108	0.822±0.096
LSR-L2	0.740±0.120	0.784±0.109	0.795±0.106	0.806±0.103	0.813±0.100	0.822±0.096
KNN	0.641±0.135	0.710±0.119	0.731±0.114	0.759±0.110	0.782±0.106	0.822±0.100
D-GEX	0.750±0.120	0.789±0.109	0.801±0.107	0.819±0.103	0.832±0.099	0.851±0.091
SemiGAN	**0.761±0.119**	**0.801±0.110**	**0.816±0.107**	**0.835±0.103**	**0.850±0.099**	**0.870±0.093**

Table 15: MAE comparison on GTEx data with different portion of labeled data.

Methods	**1%**	**3%**	**5%**	**10%**	**20%**	**100%**
LSR	2.191±0.656	0.631±0.146	0.563±0.134	0.517±0.128	0.494±0.125	0.470±0.124
LSR-L1	0.543±0.132	0.497±0.127	0.491±0.127	0.484±0.127	0.482±0.127	0.467±0.127
LSR-L2	0.519±0.118	0.490±0.121	0.487±0.121	0.482±0.123	0.478±0.123	0.468±0.123
KNN	0.676±0.137	0.653±0.147	0.650±0.145	0.638±0.147	0.632±0.147	0.623±0.147
D-GEX	0.539±0.124	0.485±0.121	0.492±0.122	0.466±0.126	0.451±0.125	0.439±0.124
SemiGAN	**0.511±0.120**	**0.475±0.123**	**0.464±0.123**	**0.447±0.124**	**0.434±0.125**	**0.422±0.127**

Table 16: CC comparison on GTEx data with different portion of labeled data.

Methods	**1%**	**3%**	**5%**	**10%**	**20%**	**100%**
LSR	0.167±0.141	0.627±0.212	0.665±0.211	0.692±0.210	0.705±0.210	0.718±0.207
LSR-L1	0.626±0.224	0.667±0.215	0.667±0.216	0.670±0.218	0.669±0.219	0.716±0.219
LSR-L2	0.613±0.220	0.677±0.212	0.687±0.212	0.700±0.211	0.707±0.211	0.718±0.208
KNN	0.462±0.214	0.521±0.213	0.529±0.211	0.551±0.209	0.560±0.208	0.575±0.205
D-GEX	0.629±0.212	0.682±0.213	0.682±0.212	0.702±0.211	0.719±0.212	0.730±0.207
SemiGAN	**0.639±0.219**	**0.693±0.214**	**0.703±0.213**	**0.721±0.212**	**0.732±0.211**	**0.744±0.209**

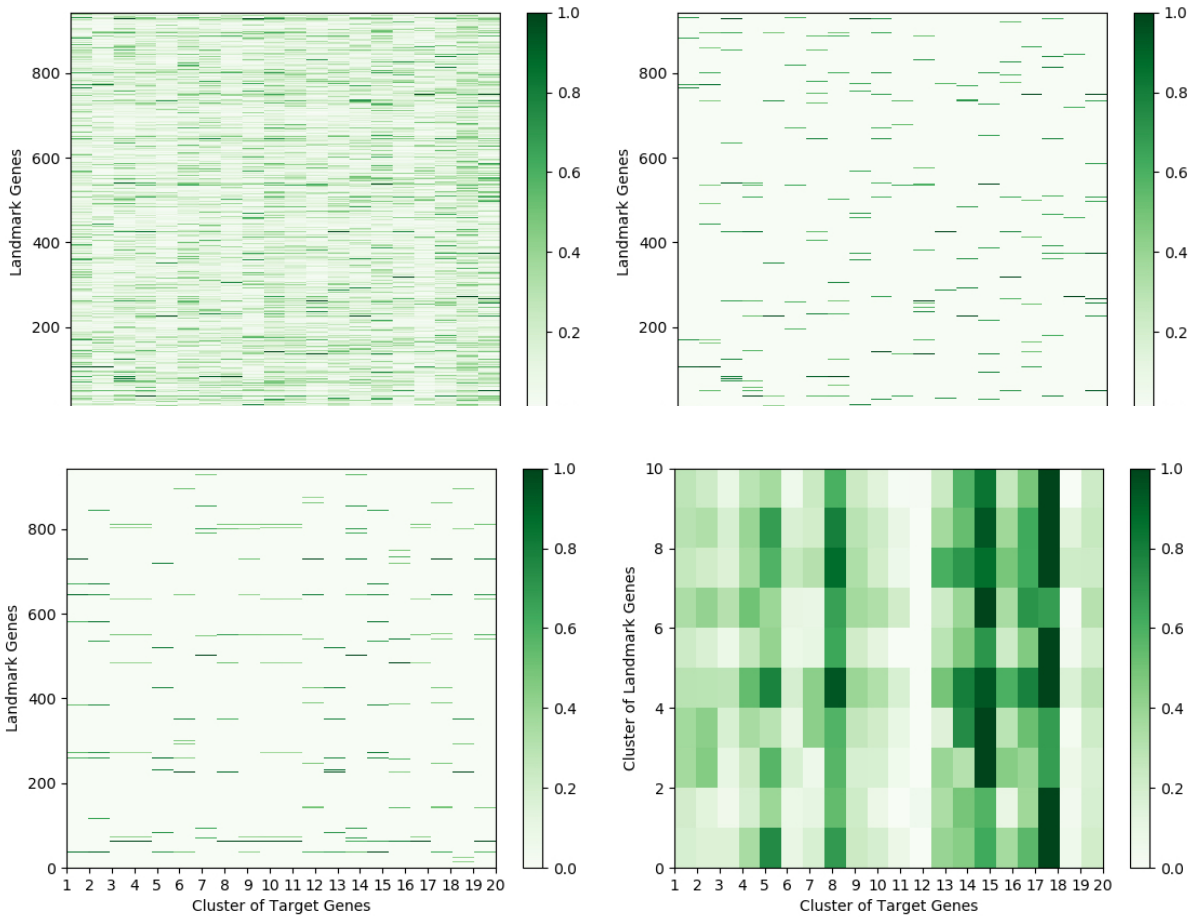

Figure 12: Visualization of the relevance score calculated by SemiGAN for each landmark gene on GEO data.

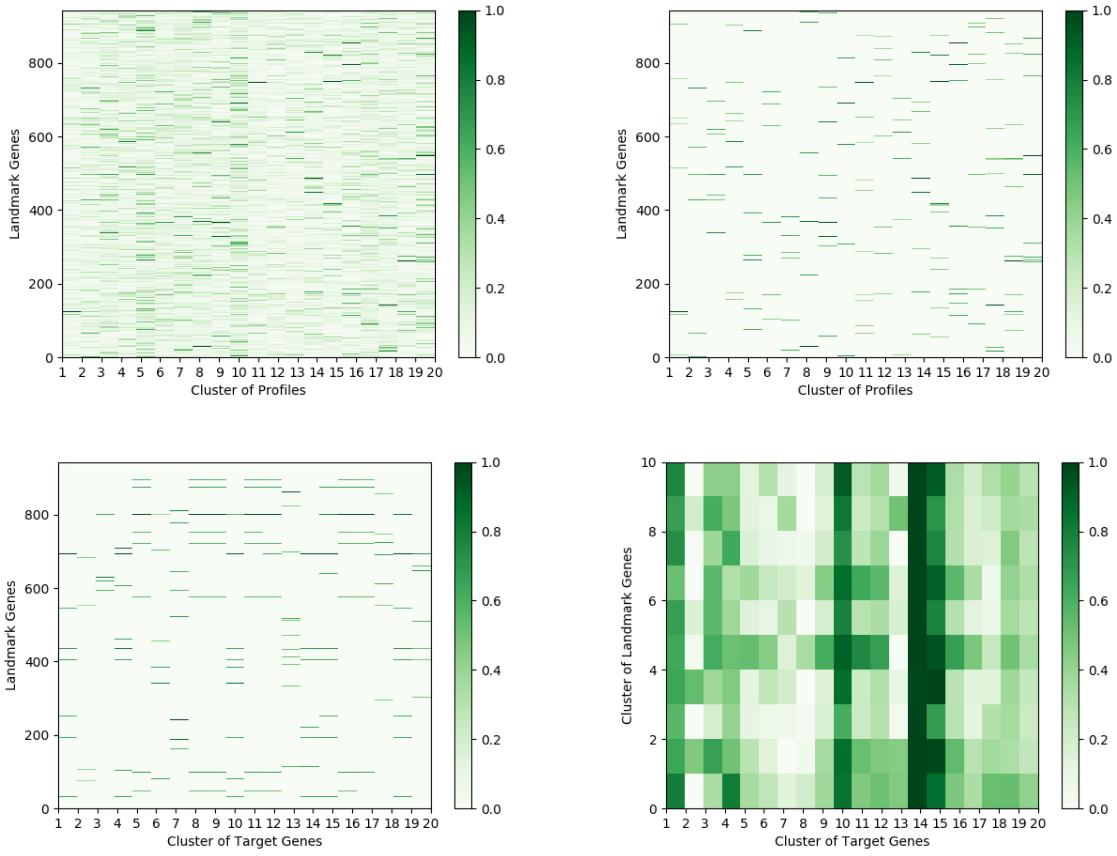

Figure 13: Visualization of the relevance score calculated by SemiGAN for each landmark gene on GTEx data.

5.0 Learning Longitudinal Data with Deep Neural Network

5.1 Motivation

As introduced in Section 1, MCI is an important intermediate stage and recent works have proposed various methods for early detecting Alzheimer's disease by distinguishing between MCI converters and non-converters. Despite the prosperity and progress achieved in MCI conversion prediction, there are still several problems existing in previous methods. 1) Although we expect the model to be capable of forecasting the MCI conversion years before the change of disease status, the training process should not be limited to just baseline data. In the longitudinal study of AD, usually the data at several time points along the disease progression is available, such as baseline, month 6, month 12, *etc.* However, previous methods only consider the baseline data in the training process, thus ignore the temporal correlation structure among other time points. 2) The labeling process for Alzheimer's is time-consuming and expensive, so the MCI conversion prediction suffers greatly from limited training data.

To deal with these problems, we propose a novel model for MCI conversion prediction. Firstly, we study the temporal correlation structure among the longitudinal data in Alzheimer's progression. Since AD is a chronically progressive disorder and the neuroimaging features are correlated [85], it can be helpful to analyze the temporal correlation between neuroimaging data in the disease progression as in other nervous system diseases [43]. We construct a regression model to discover such temporal correlation structure between adjacent time points. Our model incorporates the data at all time points along the disease progression and uncovers the variation trend that benefits MCI conversion prediction.

Secondly, we construct a classification model to predict the disease status at each time point. Different from previous classification models that use the baseline data to forecast the progression trend in two or three years, our classification model focuses on adjacent time points. Compared with previous models that require a highly distinguishable conversion pattern appears several years before dementia, our model predicts the progression trend for consecutive time points, thus is more accurate and reliable.

Thirdly, we construct a generative model based on generative adversarial network (GAN) to produce more auxiliary data to improve the training of regression and classification model. GAN model is proposed in [48], which uses the adversarial mechanism to learn the inherent data distribution and generate realistic data. We use the generative model to learn the joint distribution of neuroimaging data at consecutive time points, such that more reliable training data can be obtained to improve the prediction of MCI conversion.

5.2 Temporal Correlation Structure Learning Model

5.2.1 Problem Definition

In MCI conversion prediction, for a certain sample and a time point t, we use $\mathbf{x}_t \in \mathbb{R}^p$ to denote the neuroimaging data at time t while $\mathbf{x}_{t+1} \in \mathbb{R}^p$ for the next time point, where p is the number of imaging markers. $\mathbf{y}_t \in \mathbb{R}$ is the label showing the disease status at time t and $t+1$. Here we define three different classes for $\mathbf{y}_t$: $\mathbf{y}_t = 1$ means the sample is AD at both time t and $t+1$; $\mathbf{y}_t = 2$ shows MCI at time t while AD at time $t+1$; while $\mathbf{y}_t = 3$ indicates that the sample is MCI at both time t and $t+1$. In the prediction, given the baseline data of an MCI sample, the goal is to predict whether the MCI sample will finally convert to AD or not.

5.2.2 Revisit GAN Model

GAN model is proposed in [48], which plays an adversarial game between the generator G and discriminator D. The generator G takes a random variable $\mathbf{z}$ as the input and outputs the generated data to approximates the inherent data distribution. The discriminator D is proposed to distinguish the data $\mathbf{x}$ from the real distribution and the data produced from the generator. Whereas the generator G is optimized to generate data as realistic as possible to fool the discriminator. The objective function of the GAN model has the following form.

$$\min_G \max_D \ \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\big[\log(D(\mathbf{x}))\big] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big[\log(1 - D(G(\mathbf{z}))\big], \tag{5.1}$$

where $p(\mathbf{z})$ denotes the distribution of the random variable and $p(\mathbf{x})$ represents the distribution of real data. The min-max game played between G and D improves the learning of both the generator and discriminator, such that the model can learn the inherent data distribution and generate realistic data.

5.2.3 Illustration of Our Model

Inspired by [21], we propose to approximate the joint distribution of neuroimaging data at consecutive time points and data label $([\mathbf{x}_t, \mathbf{x}_{t+1}], \mathbf{y}_t) \sim p(\mathbf{x}, \mathbf{y})$ by considering the following:

$$\begin{aligned} \min_{G_t, G_{t+1}} \max_{D} \mathbb{E}_{([\mathbf{x}_t, \mathbf{x}_{t+1}], \mathbf{y}_t) \sim p(\mathbf{x}, \mathbf{y})} [\log(D([\mathbf{x}_t, \mathbf{x}_{t+1}], \mathbf{y}_t))] \\ + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}), \mathbf{y} \sim p(\mathbf{y})} [\log(1 - D([G_t(\mathbf{z}, \mathbf{y}), G_{t+1}(\mathbf{z}, \mathbf{y})], \mathbf{y}))] \,, \end{aligned} \tag{5.2}$$

where the generators take a random variable $\mathbf{z}$ and a pseudo label $\mathbf{y}$ as the input and output a data pair $([G_t(\mathbf{z}, \mathbf{y}), G_{t+1}(\mathbf{z}, \mathbf{y})], \mathbf{y})$ that is as realistic as possible. Still, the discriminator is optimized to distinguish real from fake data. The construction of such generative model approximates the inherent joint distribution of neuroimaging data at adjacent time points and label, which generates more reliable samples for the training process.

To uncover the temporal correlation structure among the neuroimaging data between consecutive time points, we construct a regression network R to predict $\mathbf{x}_{t+1}$ from $\mathbf{x}_t$, such that progression trend among neuroimaging data along the disease progression can be learned. The network R takes data from both real distribution and the generators as the input and optimize the following:

$$\begin{aligned} \min_{R} \mathbb{E}_{([\mathbf{x}_t, \mathbf{x}_{t+1}], \mathbf{y}_t) \sim p(\mathbf{x}, \mathbf{y})} [\|\mathbf{x}_{t+1} - R(\mathbf{x}_t)\|_1] \\ + \lambda_{reg} \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}), \mathbf{y} \sim p(\mathbf{y})} [\|G_{t+1}(\mathbf{z}, \mathbf{y}) - R(G_t(\mathbf{z}, \mathbf{y}))\|_1] \,, \end{aligned} \tag{5.3}$$

where the hyper-parameter λ_{reg} balances the importance of real and generated data. We consider ℓ_1-norm loss to make the model R more robust to outliers.

In addition, we construct a classification structure C to predict the label $\mathbf{y}_t$ given data $\mathbf{x}_t$. The optimization of C is based on the following:

$$\min_{C} \; -\mathbb{E}_{([\mathbf{x}_t, \mathbf{x}_{t+1}], \mathbf{y}_t) \sim p(\mathbf{x}, \mathbf{y})} [\mathbf{y}_t \log(C(\mathbf{x}_t))] - \; \lambda_{cly} \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z}), \mathbf{y} \sim p(\mathbf{y})} [\mathbf{y} \log(C(G_t(\mathbf{z})))] \,. \tag{5.4}$$

Given a set of real data $\{([\mathbf{x}_t^i, \mathbf{x}_{t+1}^i], \mathbf{y}_t^i)\}_{i=1}^n$, the above three loss terms can be approximated by the following empirical loss:

$$\mathcal{L}_{adv} = \frac{1}{n}\sum_{i=1}^{n} \log(D([\mathbf{x}_t^i, \mathbf{x}_{t+1}^i], \mathbf{y}_t^i)) + \sum_{j=1}^{n_z} \log(D([G_t(\mathbf{z}^j, \mathbf{y}^j), G_{t+1}(\mathbf{z}^j, \mathbf{y}^j)], \mathbf{y}^j))\,, \tag{5.5}$$

$$\mathcal{L}_{reg} = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_{t+1}^i - R(\mathbf{x}_t^i)\|_1 + \lambda_{reg}\sum_{j=1}^{n_z} \|G_{t+1}(\mathbf{z}^j, \mathbf{y}^j) - R(G_t(\mathbf{z}^j, \mathbf{y}^j))\|_1\,, \tag{5.6}$$

$$\mathcal{L}_{cly} = -\frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_t^i \log(C(\mathbf{x}_t^i)) - \lambda_{cly}\sum_{j=1}^{n_z} \mathbf{y}^j \log(C(G_t(\mathbf{z}^j, \mathbf{y}^j)))\,. \tag{5.7}$$

For a clear illustration, we plot a figure in Figure 15 to show the structure of our Temporal-GAN model (temporal correlation structure learning for MCI conversion prediction with GAN). The implement details of the networks can be found in the experimental setting section. The optimization of our model is based on a variant of mini-batch stochastic gradient descent method.

5.3 Experimental Results

5.3.1 Experimental Setting

To evaluate our Temporal-GAN model, we compare with the following methods:
SVM-Linear(support vector machine with linear kernel), which has been widely applied in MCI conversion prediction [55, 123]; **SVM-RBF** (SVM with RBF kernel), as employed in[78, 156]; and **SVM-Polynomial** (SVM with polynomial kernel) as used in [78]. Also, to validate the improvement by learning the temporal correlation structure, we compare with the **Neural Network** with exactly the same structure in our classification network (network C in Figure 15) that only uses baseline data. Besides, we compare with the case where we do not use the GAN model to generate more auxiliary samples, *i.e.,* only using network C and R in Figure 15, which we call **Temporal-Deep**.

The classification accuracy is used as the evaluation metric. We divide the data into three sets: training data for training the models, validation data for tuning hyper-parameters, and testing data for reporting the results. We tune the hyper-parameter C of SVM-linear, SVM-RBF and SVM-Polynomial methods in the range of $\{10^{-3},\ 10^{-2},\ \ldots,\ 10^{3}\}$. We compare the methods when

using different portion of testing samples and report the average performance in five repetitions of random data division. In our Temporal-GAN model, we use the fully connected neural network structure for all the networks G, D, R and C, where each hidden layer contains 100 hidden units. The implementation detail is as follows: the number of hidden layers in structure G, D, R and C is $3, 1, 3, 2$ respectively. We use leaky rectified linear unit (LReLU) [90] with leakiness ratio 0.2 as the activation function of all layers except the last layer and consider weight normalization [120] for layer normalization. Also, we utilize the dropout mechanism in the regression structure R with the dropout rate of 0.1. The weight parameters of all layers are initialized using the Xavier approach [46]. We use the ADAM algorithm [71] to update the weight parameters with the hyper-parameters of ADAM algorithm set as default. Both values of λ_{reg} in Eq. (5.3) and λ_{cly} in Eq. (5.4) are set as 0.01.

5.3.2 Data Description

Data used in this chapter was downloaded from the ADNI database (`adni.loni.usc.edu`). Each MRI T1-weighted image was first anterior commissure (AC) posterior commissure (PC) corrected using MIPAV2, intensity inhomogeneity corrected using the N3 algorithm [132], skull stripped [153] with manual editing, and cerebellum-removed [152]. We then used FAST [172] in the FSL package3 to segment the image into gray matter (GM), white matter (WM), and cerebrospinal fluid (CSF), and used HAMMER [125] to register the images to a common space. GM volumes obtained from 93 ROIs defined in [62], normalized by the total intracranial volume, were extracted as features. Out of the 93 ROIs, 24 disease-related ROIs were involved in the MCI prediction, where the selection of AD-related ROIs is based on [146]. This experiment includes six different time points: baseline (BL), month 6 (M6), month 12 (M12), month 18 (M18), month 24 (M24) and month 36 (M36). All 216 samples with no missing MRI features at BL and M36 time are used by all the comparing methods, where there are 101 MCI converters (MCI at BL time while AD at M36) as well as 115 non-converters (MCI at both BL and M36). Since Temporal-GAN model can use data at time points other than BL and M36, we include 1419 data pairs with no missing neuroimaging measurement for training the classification, regression and generative model in our Temporal-GAN model. All neuroimaging features are normalized to $\mathcal{N}(0, 1)$.

Table 17: MCI conversion prediction accuracy with different portion of testing data.

Methods	**10%**	**20%**	**50%**
SVM-Linear	0.627±0.067	0.656±0.065	0.609±0.048
SVM-RBF	0.582±0.088	0.5767±0.077	0.585±0.038
SVM-Polynomial	0.655±0.062	0.595±0.125	0.561±0.043
Neural Network	0.373±0.034	0.423±0.043	0.469±0.032
Temporal-Deep	0.746±0.046	0.721±0.044	0.674±0.045
Temporal-GAN	**0.782±0.045**	**0.749±0.034**	**0.700±0.057**

5.3.3 MCI Conversion Prediction

We summarize the MCI conversion classification results in Table 17. The goal of the experiment is to accurately distinguish converter subjects from non-converters among the MCI samples at baseline time. From the comparison we notice that Temporal-GAN outperforms all other methods under all settings, which confirms the effectiveness of our model. Compared with SVM-Linear, SVM-RBF, SVM-Polynomial and Neural Network, the Temporal-GAN and Temporal-Deep model illustrates apparent superiority, which validates that the temporal correlation structure learned in our model substantially improves the prediction of MCI conversion. The training process of our model takes advantage of all the available data along the progression of the disease, which provides more beneficial information for the prediction of MCI conversion. By comparing Temporal-GAN and Temporal-Deep, we can notice that Temporal-GAN always performs better than Temporal-Deep, which indicates that the generative structure in Temporal-GAN could provide reliable auxiliary samples to strengthen the training of regression R and classification C model, thus improves the prediction of MCI conversion.

5.3.4 Visualization of the Imaging markers

In this subsection, we use feature weight visualization figure in Figure 14 to validate if our Temporal-GAN can detect disease-related features when using all 93 ROIs in the MCI conversion

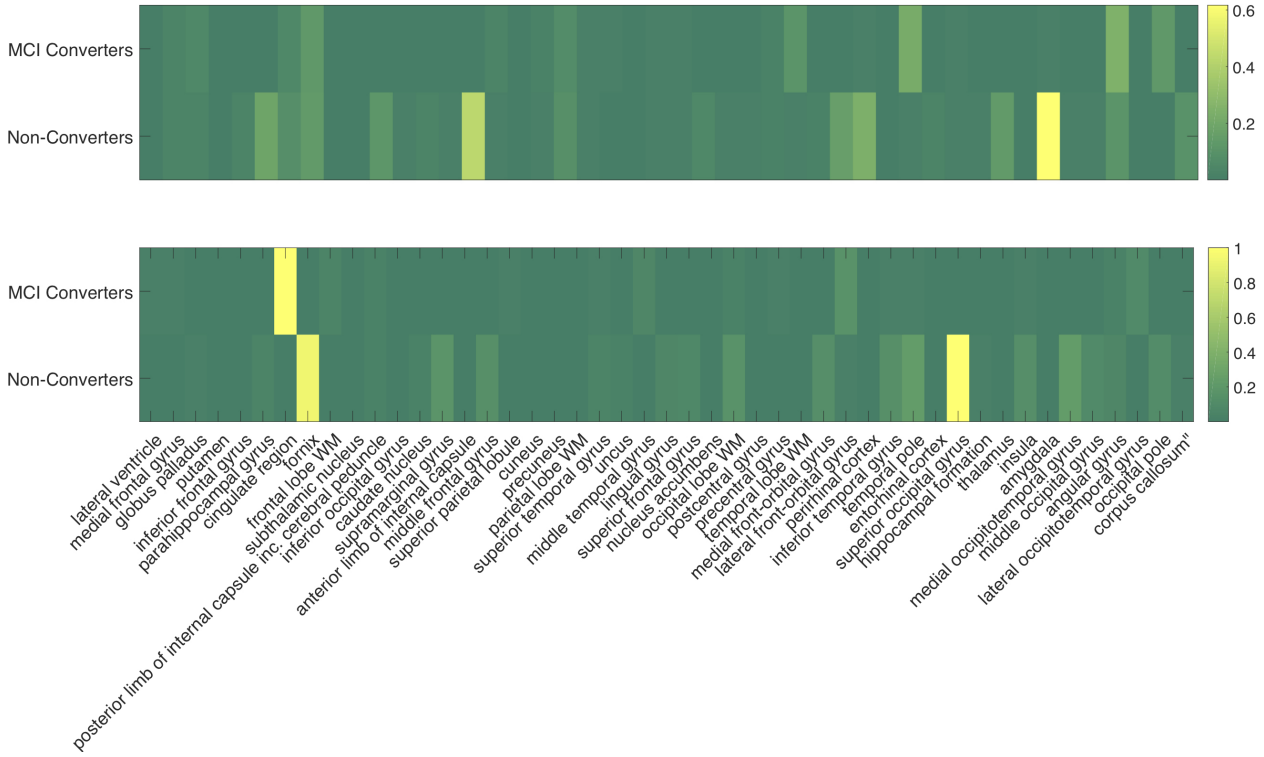

Figure 14: Visualization of the feature weights learned from Temporal-GAN.

prediction. We adopt the Layer-wise Relevance Propagation (LRP) [8] method to calculate the importance of neuroimaging features in the testing data. We can notice that our Temporal-GAN model selects several important features from all 93 ROIs. For example, our method identifies fornix as a significant feature in distinguishing MCI non-converters. The fornix is an integral white matter bundle that locates inside the medial diencephalon. [101] reveals the vital role of white matter in Alzheimer's, such that the degradation of fornix indicates essential predictive power in MCI conversion. Moreover, cingulate region has been found by our model to be related with MCI converters. Previous study [56] finds significantly decreased Regional cerebral blood flow (rCBF) measurement in the left posterior cingulate cortex in MCI converters, which serves as an important signal in forecasting the MCI conversion. The replication of these findings proves the validity of our model.

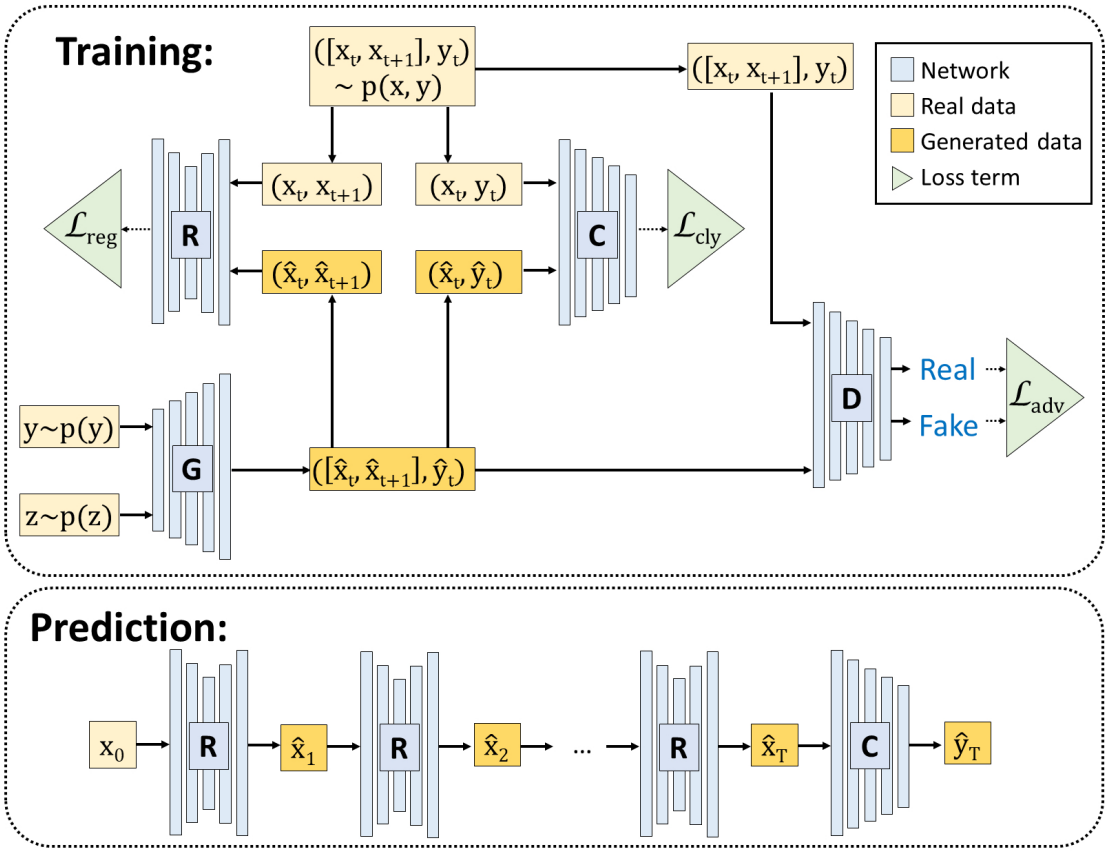

Figure 15: Illustration of our Temporal-GAN model.

6.0 Building An Additive Interpretable Deep Neural Network

6.1 Motivation

We introduce a new longitudinal deep learning model for early detection of Alzheimer's disease and improve the performance with significant margin in the previous chapter. However, the Temporal-GAN model is formulated as a black-box, making it difficult to interpret how the prediction is made and what are the important brain regions related with the occurrence and progression of the disease.

In order to address this challenge, here in this section, we propose a new idea to improve the interpretability of our model. Interpretability of the model is very important to inspect the validity of the prediction. In medical diagnosis, it is crucial to verify that the diagnosis is based on valid reasons. A thorough knowledge of the model behavior is essential to the enhance end-user's understanding and trust in the model.Moreover, good interpretability of the model can improve the performance of the model [3], strengthen the explanation of an algorithm's decision[49, 69], and enable the discovery of new science [130]. In this chapter, we formulate our deep learning model in a novel interpretable manner in order to improve the understanding of the predictive mechanism and provide insights on important features in Alzheimer's disease research.

There are several recent work on building interpretable models using sparse linear models, decision trees and rules lists owing to the intrinsic interpretable property of these methods. These models are very useful when interpretation of the computational model is important. For example, [149] propose a sparse linear model to analyze Alzheimer's disease related brain regions. [76] use the decision sets to construct an interpretable classifier for stroke prediction. [79] use rules list to build an interpretable model for stroke prediction while [145] adopt rules list to predict hospital readmissions. However, the performance of these interpretable models are limited by the complexity constraints. The number of non-zeros coefficients in sparse linear models, and the number and length of rules in rules lists and decision sets have to be carefully constrained, such that the model is not too complex to interpret. Such constrains limit the flexibility of the interpretable models, making it difficult to apply to large-scale and complex tasks.

6.1.1 Interpretation of a Black-Box Model

Recent methods on interpreting a black-box model can be roughly divided into three categories: feature visualization, attribution and model approximation. In feature visualization [38, 45, 103], the interpretation is based on finding the input that activates a certain output. From the simulated input, human users can analyze the model's understanding in the data by visualizing the desired input of the model. Moreover, attribution methods [8, 97, 168, 136, 131, 130, 4] propose to identify which part of input is the most responsible for the output by tracking from the output backward to the input. Attribution methods show the role of each feature and help the users to interpret the contribution of the features in the prediction. Model approximation is the method that builds a simpler interpretable model to approximate the black-box model and use the explanation from the simpler model for explanation. In [115], Ribeiro *et al.* build a sparse linear model to approximate any given classifier such that the output of the linear model is locally consistent to the output from the black-box classifier. In [10], Bastani *et al.* use decision trees to approximate the black-box model for a global interpretation of the model behavior. Also, there are other model interpretation methods in addition to the above three categories. [73] formulates the interpretation by analyzing the impact of each training sample to the model and picking out the most influential training samples in the prediction.

Although the recent works have introduced advances in model interpretation and strengthened the understanding in black-box models, there still lacks a universal standard in evaluating and quantifying the quality of interpretation [33]. [70] points out that several well-known interpretation methods, including DeConvNet [168], Guided BackProp [136] and LRP [8], fail to provide theoretically correct interpretation for a simple linear model. In [1], Adebayo *et al.* points out that a deep neural network (DNN) with random weights shares both visual and quantitative similarity with a DNN with learned weights on the learned interpretation. All of these findings raise concerns about the quality and consistency of interpretation. In [24], Chu *et al.* propose a theoretical analysis on the consistency of the interpretation, whose method is, however, constrained to piecewise linear models and is not applicable to neural networks with other standard layers like batch normalization, *etc.*

It is notable that our interpretable temporal structure learning model is different from the previous works from several different aspects. In terms of MCI conversion prediction, we propose a novel idea of uncovering the progressive trend of Alzheimer's disease to improve the prediction of MCI conversion status. Moreover, unlike previous models that predict the MCI conversion two or three years before the disease status changes, our prediction focuses on adjacent time points thus is more accurate.

In terms of model interpretation, our model differs from the model interpretation methods in that we directly construct an interpretable model, guaranteeing that the interpretation is consistent with the model behavior. In addition, different from previous interpretable models whose performance is limited by the interpretable constraints, our model is flexible and achieves roughly comparable performance with state-of-the-art black-box structures on difficult tasks such as image classification and sentence classification.

6.2 Building An Additive Interpretable Deep Neural Network

In this section, we first introduce how to build a deep learning model in an interpretable manner such that the model provides direct explanation on what features play an important role in the prediction. In a supervised learning problem, suppose the data is drawn from the distribution $p(\mathbf{x}, \mathbf{y})$, traditional black-box models propose to find a prediction function f that optimizes the following:

$$\min_{f \in \mathcal{F}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})} \Big[\mathcal{L}(f(\mathbf{x}), \mathbf{y}) \Big], \tag{6.1}$$

where $\mathcal{L}$ is the loss function and $\mathcal{F}$ is the assumptions and constraints on the structure of f. In deep learning models, function f is non-linear and complicated, which makes the behavior of function f difficult to understand. For a given data $\mathbf{x}$, such black-box structure renders the mechanism behind how the model makes the prediction $\hat{\mathbf{y}} = f(\mathbf{x})$ to be opaque and not interpretable.

One method to understand the behavior of f is to find an interpretable model h (*e.g,* sparse linear model $h(\mathbf{x}) = \mathbf{x}^\mathsf{T} W$) to locally approximate f [115]. It is notable that the coefficient matrix W is dependent on the data $\mathbf{x}$ to be interpreted. As a consequence, we can define W as a function *w.r.t.* the data $\mathbf{x}$ as $W = g(\mathbf{x})$ and formulate the prediction as $\hat{\mathbf{y}} = \mathbf{x}^\mathsf{T} g(\mathbf{x})$. We plug $\hat{\mathbf{y}} = \mathbf{x}^\mathsf{T} g(\mathbf{x})$ in

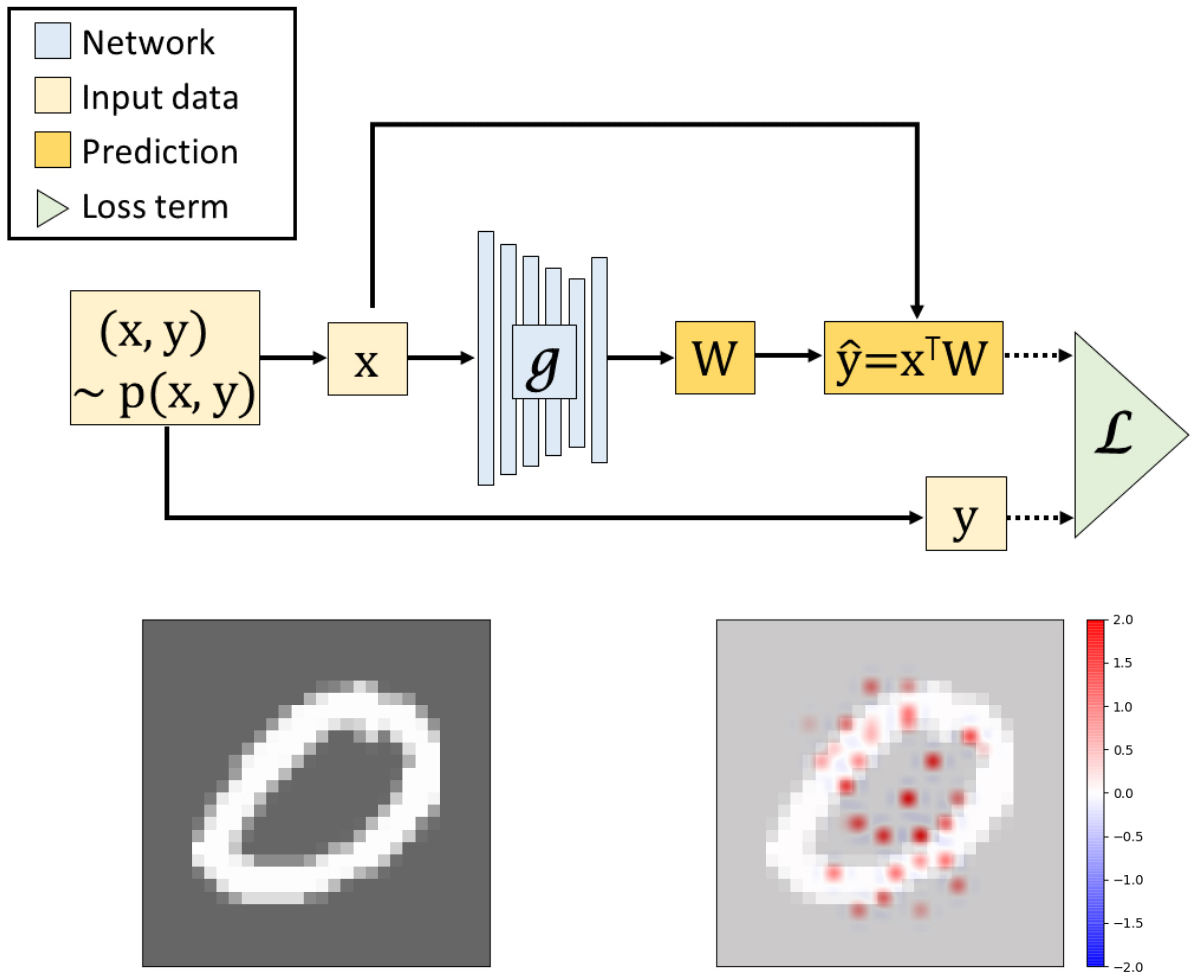

Figure 16: Illustration of the idea of constructing an interpretable additive deep neural network, with the illustrating example shown below.

Eq. (6.1) and propose to optimize the following:

$$\min_{g \in \mathcal{G}} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})} \left[\mathcal{L}(\mathbf{x}^\mathsf{T} g(\mathbf{x}), \mathbf{y}) \right]. \tag{6.2}$$

where $\mathcal{G}$ is the assumptions and constraints on the structure of function g. Note that in Problem (6.2), $g(\mathbf{x})$ makes an explicit interpretation on the contribution of features in $\mathbf{x}$ to $\hat{\mathbf{y}}$, such that the mechanism behind the model behavior is clear.

Given a set of data samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, we optimize the following empirical loss to approximate the expected loss in Problem (6.2):

$$\min_{g \in \mathcal{G}} \sum_{i=1}^{n} \mathcal{L}(\mathbf{x}_i^\mathsf{T} g(\mathbf{x}_i), \mathbf{y}_i) \,. \tag{6.3}$$

For the sake of interpretability, we impose sparsity constraints on the learned coefficient matrix to limit the number of non-zero values in the coefficient matrix for each data and propose the following objective function for our method:

$$\min_{g \in \mathcal{G}} \sum_{i=1}^{n} \mathcal{L}(\mathbf{x}_i^\mathsf{T} g(\mathbf{x}_i), \mathbf{y}_i) \,, \;\; s.t. \;\; \|g(\mathbf{x}_i)\|_{2,0} \leq k, \;\; \forall i \,, \tag{6.4}$$

where k is the number of non-zeros values in each row of the coefficient matrix.

We would like to point out that the formulation in Problem (6.4) enjoys several advantages:

- The model is clearly interpretable since the coefficient matrix $g(\mathbf{x}_i)$ makes an explicit and direct explanation on the association between data $\mathbf{x}_i$ and the prediction $\hat{\mathbf{y}}_i = \mathbf{x}_i^\mathsf{T} g(\mathbf{x}_i)$.
- The function g enjoys the representative power and flexibility of the black-box models, thus our model can perform well in the cases when f works well in Problem (6.1).
- Our interpretable learning idea is not restricted to any specific structure and can adapt to many state-of-the-art architecture.

For simplicity, we plot an illustration figure in Figure 16 to explain our idea of building an interpretable model. The learned coefficient matrix on MNIST data for digit classification indicates that our model correctly classifies the digit according to reasonable features, showing that our model is both effective and trustworthy.

We formulate the classifier in the Temporal-GAN model introduced in Chapter 5 and construct ITGAN (interpretable temporal correlation structure learning for MCI conversion prediction with GAN). The optimization of our ITGAN model is based on a variant of mini-batch stochastic gradient descent method.

6.3 Experimental Results

6.3.1 Experimental Setting

To evaluate our ITGAN model, we compare with the following methods:
SVM-Linear(support vector machine with linear kernel), which has been widely applied in MCI conversion prediction [55, 123]; **SVM-RBF** (SVM with RBF kernel), as employed in[78, 156]; and **SVM-Polynomial** (SVM with polynomial kernel) as used in [78]. Also, to validate the improvement by learning the temporal correlation structure, we compare with the **Neural Network** with exactly the same structure in our classification network (network C in Figure 15) that only uses baseline data. Besides, we compare with the case where we do not use the GAN model to generate more auxiliary samples, *i.e.,* only using network C and R in Figure 15, which we call **Temporal-Deep**.

The classification accuracy is used as the evaluation metric. We divide the data into three sets: training data for training the models, validation data for tuning hyper-parameters, and testing data for reporting the results. We tune the hyper-parameter C of SVM-linear, SVM-RBF and SVM-Polynomial methods in the range of $\{10^{-3},\ 10^{-2},\ \ldots,\ 10^{3}\}$. We compare the methods when using different portion of testing samples and report the average performance in five repetitions of random data division. It is notable that our classification network C output the weight matrices W such that the prediction of C is intrinsically interpretable. We plot an illustration figure in Figure 17.

In our ITGAN model, we use the fully connected neural network structure for all the networks G, D, R and C. The implementation detail is as follows: The number of each hidden layer contains 100 hidden units while the number of hidden layers in the structure G, D, R and C is $2, 1, 3, 4$ respectively. We use leaky rectified linear unit (LReLU) [90] with leakiness ratio 0.2 as the activation function of all layers except the last layer and consider weight normalization [120] for layer normalization. We adopt the dropout mechanism for all layers except the last layer with the dropout rate of 0.1 [137]. The weight parameters of all layers are initialized using the Xavier approach [46]. We use the ADAM algorithm [71] to update the weight parameters with the hyper-parameters of ADAM algorithm set as default. Both values of λ_{reg} in Eq. (5.3) and λ_{cly} in Eq.

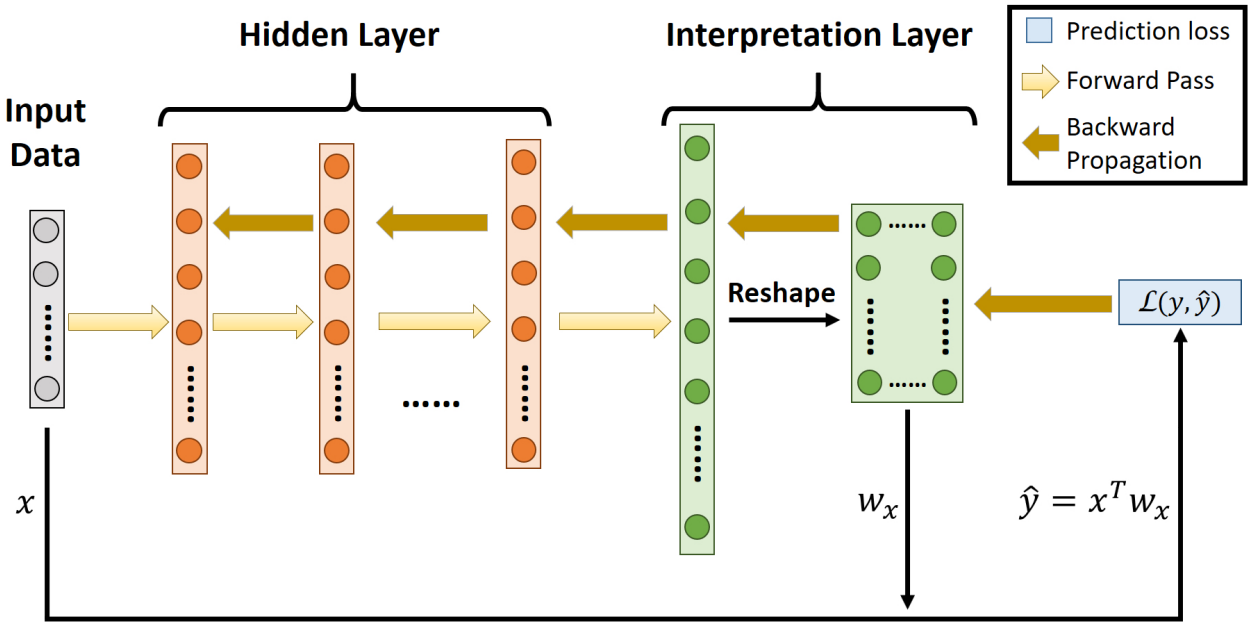

Figure 17: Illustration of our idea on formulating a deep learning model in an interpretable manner.

(5.4) are set as 1.0. We use the generator G to produce ten times of the samples from the real distribution. We use Theano toolbox for the implementation. All experiments are conducted on a machine with one Titan X pascal GPU.

6.3.2 MCI Conversion Prediction

We summarize the MCI conversion classification results in Table 18 and 19. The goal of the experiment is to accurately distinguish converter subjects from non-converters among the MCI samples at baseline time. From the comparison we notice that ITGAN outperforms all other methods under all settings, which confirms the effectiveness of our model. Compared with SVM-Linear, SVM-RBF, SVM-Polynomial and Neural Network, the ITGAN and Temporal-Deep model illustrates apparent superiority, which validates that the temporal correlation structure learned in our model substantially improves the prediction of MCI conversion. The training process of our model takes advantage of all the available data along the progression of the disease, which provides more

Table 18: Classification evaluation of ITGAN on MCI conversion prediction with different portion of testing data.

Methods	**10%**	**20%**	**50%**
SVM-Linear	0.627±0.067	0.656±0.065	0.609±0.048
SVM-RBF	0.582±0.088	0.577±0.077	0.585±0.038
SVM-Polynomial	0.655±0.062	0.595±0.125	0.561±0.043
Neural Network	0.373±0.034	0.423±0.043	0.469±0.032
Temporal-Deep	0.746±0.068	0.744±0.042	0.756±0.036
ITGAN	**0.809±0.088**	**0.749±0.027**	**0.761±0.030**

Table 19: Classification evaluation of ITGAN when involving all 93 ROIs in MCI conversion prediction.

Methods	**10%**	**20%**	**50%**
SVM-Linear	0.618±0.121	0.614±0.032	0.582±0.052
SVM-RBF	0.673±0.116	0.558±0.051	0.606±0.040
SVM-Polynomial	0.618±0.162	0.656±0.103	0.598±0.047
Neural Network	0.446±0.105	0.461±0.045	0.459±0.013
Temporal-Deep	0.773±0.050	0.781±0.048	0.700±0.031
ITGAN	**0.809±0.067**	**0.786±0.037**	**0.715±0.042**

beneficial information for the prediction of MCI conversion. By comparing ITGAN and Temporal-Deep, we can notice that ITGAN always performs better than Temporal-Deep, which indicates that the generative structure in ITGAN could provide reliable auxiliary samples to strengthen the training of regression R and classification C model, thus improves the prediction of MCI conversion.

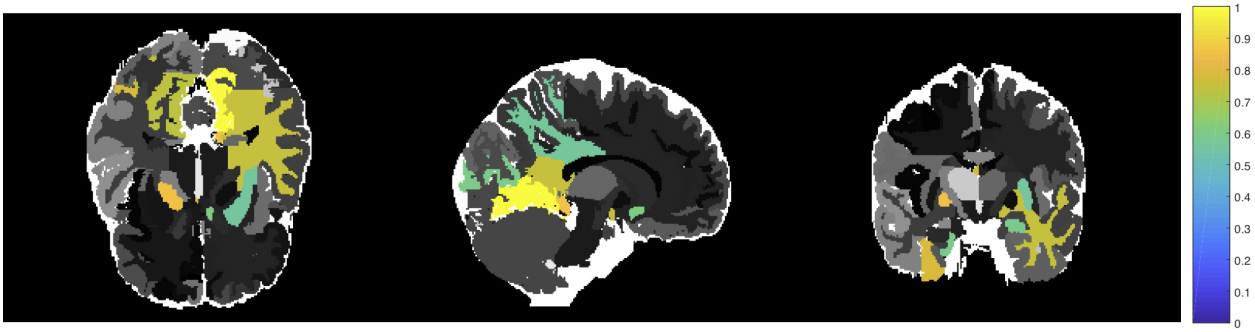

Figure 18: Brain map of the top imaging markers identified by ITGAN.

6.3.3 Visualization of the Imaging markers

In Figure 18, we illustrate the brain map corresponding to the top 15 imaging features that are learned in our C model. In this subsection, we use all 93 ROIs in the MCI conversion prediction and validate if the top neuroimaging features captured by our ITGAN model is Alzheimer's disease relevant.

Among the detected neuroimaging features, we identified several Alzheimer's relevant regions of interests (ROIs) that have been replicated in previous literature. In our classification C model, medial occipitotemporal gyrus left and inferior temporal gyrus right rank play an important role in the early detection of MCI. Convit *et al.* [26] ascertained the influence of the atrophy in the medial occipitotemporal, inferior, and middle temporal gyri in the decline to Alzheimer's disease. Moreover, our ITGAN model also validated temporal lobe WM left as a top feature to predict the conversion of MCI, the this finding has been replicated in several previous works [66, 16]. The replication of this findings validated the effectiveness of our interpretation method, such that our ITGAN model provides direct explanation of the important features for predicting the conversion pattern of MCI subjects, and the explanation coincides with previous research on the important features for MCI subjects declining to Alzheimer's disease.

6.4 Additive Interpretation Methods for Machine Learning Fairness

In recent years, machine learning has achieved unparalleled success in various fields, from face recognition, autonomous driving to computer-aided diagnosis. Despite the wide application and rapid development, the discrimination and bias that exists in machine learning models are attracting increasing attention in the research community. Recent models have been found to be biased towards certain groups of samples when making the prediction. For example, ProPublica [61] analyzed a widely used criminal risk assessment tool for future crime prediction and discovered discrimination among different races. For defendants that do not commit a future crime, the black people are more likely to be mistaken by the model as potential future criminals than the white people (*i.e.,* a higher false positive rate in the blacks than the whites). Moreover, Gross *et al.* [50] analyzed the face recognition problem and uncovered prediction discrimination among ethnicity, such that the recognition accuracy of white people is much higher than that of the black people.

Especially in sensitive fields such as criminal justice, credit and loan decision, and online advertising, a model with merely good prediction performance is not enough as we harness the power of machine learning. It is critical to guarantee that the prediction is based on appropriate information and the performance is not biased towards certain groups of population characterized by sensitive features like race and gender.

Improving model fairness is not only a societal problem but also an important aspect of machine learning. As the prediction bias uncovered in various applications, there are rising concerns *w.r.t.* the discrimination among sensitive groups and thus the trustworthiness of model performance. Recent works propose to achieve machine learning fairness from different perspectives to improve model fairness. For example, as a pre-processing step, recent methods propose to eliminate the bias in data with reweighing the samples [63] or removing the disparity among groups [41]. While in the in-processing of model prediction, Zhang *et al.* [170] proposes to improve fairness by constraining the prediction not based on sensitive information. Adel *et al.* [2] also propose an adversarial network that minimizes the influence of sensitive features to the prediction by characterizing the relevance between the latent data representation and the sensitive feature. Besides, fairness in prediction can be achieved with post-processing methods [110] that modifies the model output for equalizing the probability of getting favorable output, *e.g.,* getting approved for a loan.

Based on the targets of fairness, the motivation can be divided into group fairness and individual fairness. Group fairness is proposed to guarantee that different groups of population have equalized opportunity of achieving a favorable prediction result. Whereas for individual fairness [169], the goal is to guarantee that similar individuals get similar output. Based on the motivation of improving fairness, there are recent methods proposed to improve the long-term benefit of the protected groups (groups that are usually biased against by traditional models) [84, 99], which is different than the methods that focus more on the instant benefit of an equalized opportunity [110].

Previous models usually propose to improve the fairness *w.r.t.* either the data perspective or the model perspective, *i.e.,* modifying the input to reduce data bias or optimizing the model to reduce prediction bias. These strategies may not guarantee that the learned input to be optimal for the model or the designed model optimal for the data, such that a fairness constraint in the model usually introduces deterioration in the prediction performance.

In order to improve fairness without sacrificing the predictive performance, we propose a new adversarial network to reduce the bias simultaneously from the data perspective and the model perspective. By conducting sampling among features, we automatically reformulate the input with features that contain only non-sensitive information. By minimizing the marginal contribution of the sensitive feature, we strengthen model robustness towards the sensitive feature such that adding sensitive information cannot influence the prediction results. The coupled optimization strategy from both the data and the model aspects improves fairness as well as prediction performance. We evaluate our model on three benchmark datasets, where our model achieves the best prediction performance as well as the most improved prediction fairness when compared with four state-of-art fairness models and the baseline.

6.4.1 Problem Definition

First we introduce several terminologies in machine learning fairness. For a given dataset $[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(n)}]$ consisting of n samples from the input space $\mathcal{X} \subset \mathbb{R}^d$, each sample $\mathbf{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_d^{(i)}]^\mathsf{T}$ is characterized by d features.

In a prediction problem, **prediction bias** sometime exists when the model makes different prediction for different groups of samples with all other features held constant. For example, the

Home Mortgage Disclosure Act (HMDA) data shows the rate of loan rejection is twice as high for blacks as for whites [75]. The **sensitive feature** is the feature to characterize such **groups of population** of interest where we expect the prediction not to be biased among the groups. Examples of the sensitive feature include *race, gender, age.* The choice of sensitive features varies for different prediction problems. The **sensitive-relevant features** refers to the features that are not regarded as sensitive themselves, but indicate the information relevant to the sensitive feature. For example, in a job hiring decision model, the *university* where the candidates graduate from is not a sensitive feature. However, *university* can be relevant to the sensitive feature *race* since the demographics of different universities is different.

One straightforward idea to improve fairness is **fairness through blindness**, *i.e.,* simply exclude the sensitive feature from the input. However, this cannot eliminate the prediction bias, as the sensitive-relevant features still provide sensitive information in the input.

The goal of fairness varies in different applications, such as group/individual fairness, the long-term/instant benefit of fairness as introduced in Section 6.4. Here in this work, we are interested in improving the fairness with instant benefit among different groups of population so that the model prediction is not based on the sensitive information, either from the sensitive or sensitive-relevant features.

In this chapter, we propose to reduce such prediction bias from two aspects: reformulating the *input* and strengthening the *model* fairness. We achieve the goal by simultaneously learn a new input $\tilde{\mathbf{x}}$ based on the original data $\mathbf{x}$ and build a prediction model $f^{\phi} : \mathcal{X} \rightarrow \mathcal{Y}$ with the parameter ϕ, where $\mathcal{Y}$ is the output space, such that 1) the dependency between $\tilde{\mathbf{x}}$ and the sensitive information is minimized; 2) the influence of the sensitive information to the prediction of f^{ϕ} is minimized. By improving from both the input and the model, we propose to guarantee that the prediction is based on the non-sensitive information and the bias *w.r.t.* the sensitive feature is reduced.

6.5 Approaching Machine Learning Fairness Through Adversarial Network

As we discussed in Section 6.4.1, the simple strategy of fairness through blindness cannot work with the existence of sensitive-relevant features. In order to reduce the prediction bias, we need

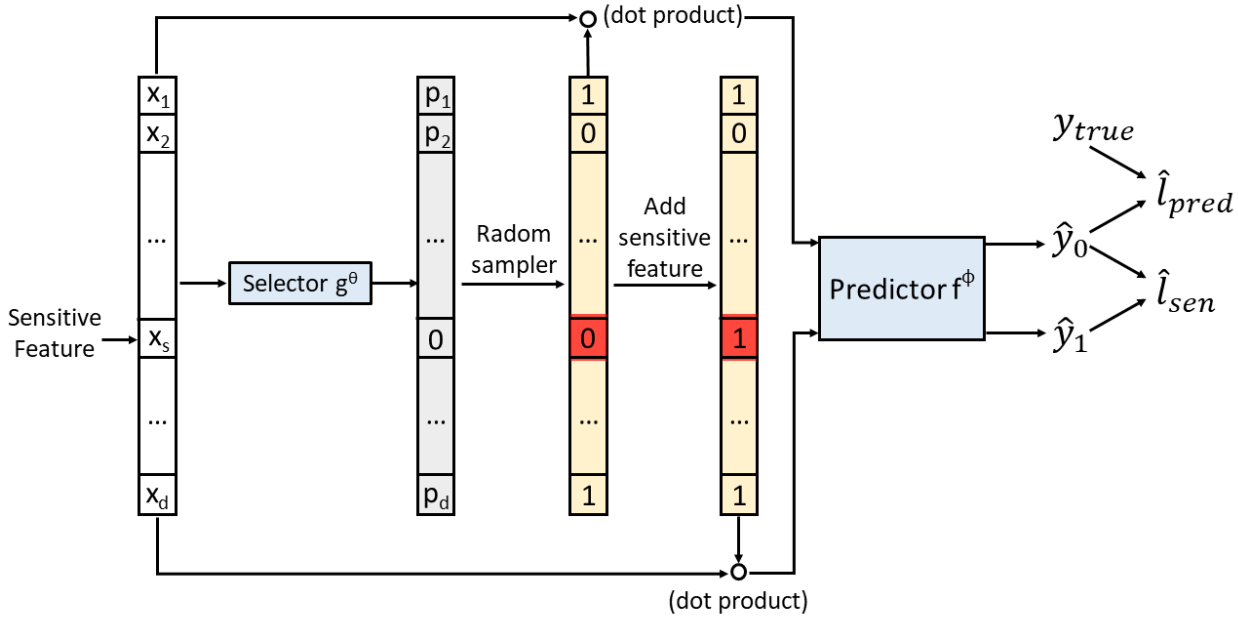

Figure 19: Illustration of the FAIAS model.

to guarantee the prediction is not dependent on either the sensitive feature or the sensitive-relevant features. This is difficult to achieve since we usually do not have prior knowledge of what are the sensitive-relevant features. In this section, we propose a new FAIrness through AdverSarial network (FAIAS) model to improve the prediction fairness by improving both the input and the model.

The goal of reducing the prediction bias from both the input and model aspects can be formulated as two folds: 1) from the perspective of input, we propose to learn the new input $\tilde{\mathbf{x}}$ based on the original data $\mathbf{x}$ such that $\tilde{\mathbf{x}}$ contains only non-sensitive information; 2) for the prediction model, we minimize the marginal contribution of the sensitive feature such that adding the sensitive feature does not change the model prediction too much. We propose to learn the new input $\tilde{\mathbf{x}}$ by sampling the features in the original data $\mathbf{x}$, *i.e.,* selecting features with a selection function $S : \mathcal{X} \rightarrow \{0, 1\}^d$, such that the selected features contain only non-sensitive information.

For a data sample $\mathbf{x} = [x_1,\ x_2,\ \ldots,\ x_d]^\mathsf{T} \in \mathcal{X}$, and a selection set $\mathbf{s} = \{s_1,\ s_2,\ \ldots,\ s_m\} \subset \{1,\ 2,\ \ldots,\ d\}$, denote $f^\phi(\mathbf{x}, \mathbf{s}) = f^\phi([x_{s_1},\ x_{s_2},\ \ldots,\ x_{s_m}])$ as the output of function f^ϕ when the input contains only features selected by $\mathbf{s}$ (the value of not selected features as set as 0). For $t \notin \mathbf{s}$, the marginal contribution of the t-th feature to this input can be denoted as $f^\phi(\mathbf{x}, \mathbf{s} \cup \{t\}) - f^\phi(\mathbf{x}, \mathbf{s})$, *i.e.,* the change in the output when adding t-th feature.

Denote the sensitive feature as x_k, the goal of our FAIAS model is to minimize the marginal contribution

$$f^\phi(\mathbf{x}, S \cup \{k\}) - f^\phi(\mathbf{x}, S),$$

where S is the selection function that selects only features containing non-sensitive information. For simplicity, here we only consider one sensitive feature in each data. It is notable that our FAIAS model can be easily applied to improving prediction fairness in the case involving multiple sensitive features.

We can approximate the selection function S using a continuous selector function $g^\theta : \mathcal{X} \to [0, 1]^d$ with parameter θ, that takes the feature vector as the input and output a probability vector $\mathbf{p} = [p_1,\ p_2,\ \ldots,\ p_d] \in \mathbb{R}^d$ showing the probability to sample each feature to formulate the input. Then we conduct random sampling of the features based on the probability vector $\mathbf{p}$ and get the selection set $\mathbf{s}$. The probability of getting a joint selection vector $\mathbf{s} \in \{0, 1\}^d$ is

$$\pi_\theta(\mathbf{x}, \mathbf{s}) = \Pi_{j=1}^d \big(g_j^\theta(\mathbf{x})\big)^{s_j} \big(1 - g_j^\theta(\mathbf{x})\big)^{(1-s_j)}. \tag{6.5}$$

To quantify the influence of sensitive feature, we consider the sensitivity loss as follows

$$l_{sen}(\theta, \phi) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p} \mathbb{E}_{\mathbf{s}\sim\pi_\theta(\mathbf{x},\cdot)} \Big[|| f^\phi(\mathbf{x}, \mathbf{s} \cup \{k\}) - f^\phi(\mathbf{x}, \mathbf{s}) || \Big], \tag{6.6}$$

which characterize the marginal contribution of the sensitive feature x_k to the model prediction given features selected by $\mathbf{s}$.

In order to optimize g^θ to approximate the selection function S and assign higher probability to only non-sensitive features, we propose an adversarial game between the selector function g^θ and the predictor function f^ϕ.

The goal of the prediction function f^ϕ is to minimize the sensitivity loss in Eq. (6.6) such that adding the sensitive feature does not influence the prediction too much. In contrast, we optimize the selector function g^θ to maximize the sensitivity loss in Eq. (6.6), so as to select the subset of

features which can be influenced the most by adding the sensitive feature. In this way, the selector function g^θ can find the features that are not relevant to the sensitive feature. If for example, the selected subset contains sensitive-relevant features, adding the sensitive feature will not bring too much change since the sensitive information is already indicated by the sensitive-relevant features. By updating the selector function g^θ to maximize the sensitivity loss, g^θ learns to exclude the sensitive information by assigning lower sampling probability to sensitive-relevant features and formulate the input on the basis of only non-sensitive information.

Moreover, we optimize the predictor f^ϕ to minimize the following prediction loss to guarantee prediction performance:

$$l_{pred}(\theta, \phi) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p}\mathbb{E}_{\mathbf{s}\sim\pi_\theta(\mathbf{x},\cdot)}\big[\sum_{l=1}^{c} y_l \log f_l^\phi(\mathbf{x}, \mathbf{s})\big], \tag{6.7}$$

which measures the performance of the prediction model given the features selected by **s**. Here we take the the multi-class classification problem with c class as an example and consider the cross entropy loss. We plot an illustration figure in Figure 6.5 to show our FAIrness through AdverSarial network (FAIAS) model.

In order to optimize the selector function g^θ and the prediction function f^ϕ, we derive the update steps for the two functions in the following.

Denote the empirical loss *w.r.t* data **x** and selection vector **s** as below:

$$\hat{l}_{sen}(\mathbf{x}, \mathbf{s}) = f^\phi(\mathbf{x}, \mathbf{s} \cup \{k\}) - f^\phi(\mathbf{x}, \mathbf{s}), \tag{6.8}$$

and the empirical loss as:

$$\hat{l}_{pred}(\mathbf{x}, \mathbf{s}) = \sum_{l=1}^{c} y_l \log f_l^\phi(\mathbf{x}, \mathbf{s}). \tag{6.9}$$

The parameter θ and ϕ can be updated via gradient ascend and descent methods respectively. We can easily derive the derivative of $l_{sen}(\theta, \phi)$ *w.r.t.* parameter θ and ϕ as follows:

$$\begin{aligned}
\nabla_\theta l_{sen}(\theta,\phi) &= \nabla_\theta \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p}\mathbb{E}_{\mathbf{s}\sim\pi_\theta(\mathbf{x},\cdot)}\Big[||f^\phi(\mathbf{x},\mathbf{s}\cup\{k\}) - f^\phi(\mathbf{x},\mathbf{s})||\Big] \\
&= \nabla_\theta \int_{\mathcal{X}\times\mathcal{Y}} p(\mathbf{x},\mathbf{y})\Big(\sum_{\mathbf{s}\in\{0,1\}^d} \pi_\theta(\mathbf{x},\mathbf{s})||\hat{l}_{sen}(\mathbf{x},\mathbf{s})||\Big)dxdy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} p(\mathbf{x},\mathbf{y})\Big(\sum_{\mathbf{s}\in\{0,1\}^d} \pi_\theta(\mathbf{x},\mathbf{s})\frac{\nabla_\theta\pi_\theta(\mathbf{x},\mathbf{s})}{\pi_\theta(\mathbf{x},\mathbf{s})}||\hat{l}_{sen}(\mathbf{x},\mathbf{s})||\Big)dxdy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} p(\mathbf{x},\mathbf{y})\Big(\sum_{\mathbf{s}\in\{0,1\}^d} \pi_\theta(\mathbf{x},\mathbf{s})\nabla_\theta\log\pi_\theta(\mathbf{x},\mathbf{s})||\hat{l}_{sen}(\mathbf{x},\mathbf{s})||\Big)dxdy \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p}\mathbb{E}_{\mathbf{s}\sim\pi_\theta(\mathbf{x},\cdot)}\Big[||\hat{l}_{sen}(\mathbf{x},\mathbf{s})||\nabla_\theta\log\pi_\theta(\mathbf{x},\mathbf{s})\Big],
\end{aligned} \tag{6.10}$$

$$\begin{aligned}
\nabla_\phi l_{sen}(\theta,\phi) &= \nabla_\phi \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p}\mathbb{E}_{\mathbf{s}\sim\pi_\theta(\mathbf{x},\cdot)}\big[||f^\phi(\mathbf{x},\mathbf{s}\cup\{k\}) - f^\phi(\mathbf{x},\mathbf{s})||\big] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p}\mathbb{E}_{\mathbf{s}\sim\pi_\theta(\mathbf{x},\cdot)}\Big[\frac{(f^\phi(\mathbf{x},\mathbf{s}\cup\{k\}) - f^\phi(\mathbf{x},\mathbf{s}))(\nabla_\phi f^\phi(\mathbf{x},\mathbf{s}\cup\{k\}) - \nabla_\phi f^\phi(\mathbf{x},\mathbf{s}))}{||\hat{l}_{sen}(\mathbf{x},\mathbf{s})||}\Big],
\end{aligned} \tag{6.11}$$

and the derivative of $l_{pred}(\theta, \phi)$ *w.r.t.* ϕ is

$$\begin{aligned}
\nabla_\phi l_{pred}(\theta,\phi) &= \nabla_\phi \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p}\mathbb{E}_{\mathbf{s}\sim\pi_\theta(\mathbf{x},\cdot)}\big[\sum_{l=1}^{c} y_l \log f_l^\phi(\mathbf{x},\mathbf{s})\big] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p}\mathbb{E}_{\mathbf{s}\sim\pi_\theta(\mathbf{x},\cdot)}\big[\sum_{l=1}^{c} y_l \frac{\nabla_\phi f_l^\phi(\mathbf{x},\mathbf{s})}{f_l^\phi(\mathbf{x},\mathbf{s})}\big].
\end{aligned} \tag{6.12}$$

In Algorithm 4 we summarize the optimization steps of FAIAS model. According to the update rules *w.r.t.* the gradients, the time complexity of our FAIAS model is linear *w.r.t.* the number of samples n, the number of parameters in θ and ϕ, as well as the number of iterations T.

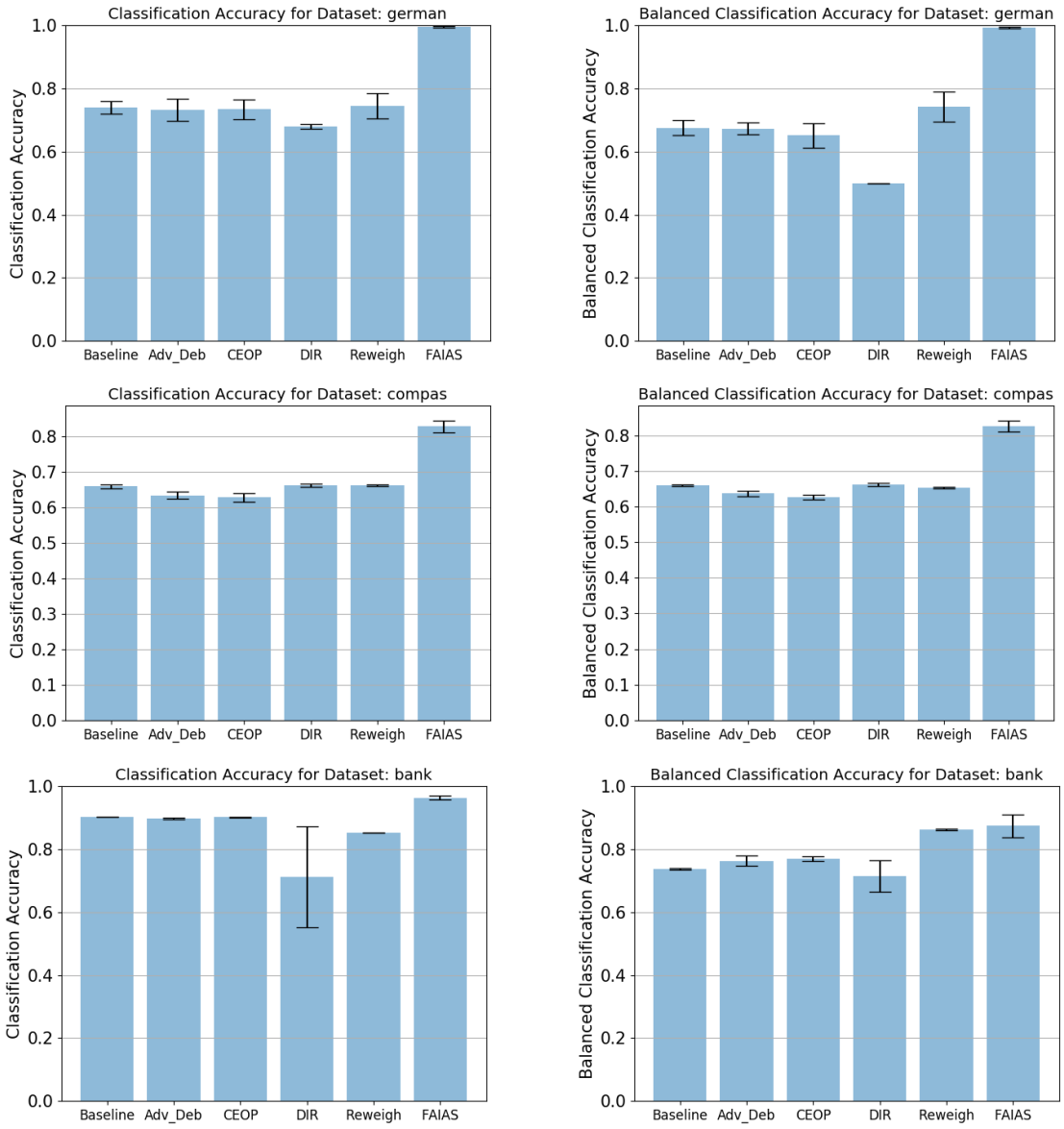

Figure 20: Comparison of model performance via classification accuracy and balanced classification accuracy on three benchmark datasets.

6.6 Experimental Results

In this section, we conduct experiments on three benchmark datasets to validate the performance of our FAIAS model. It is notable that our FAIAS model is proposed for group fairness,

i.e., minimizing the prediction bias *w.r.t.* a certain sensitive feature in both the pre-processing and in-processing steps. We compare our model with four recent methods for group fairness in pre-processing, in-processing, and post-processing steps, and one baseline method as follows.

- **Baseline method without fairness constraint**: the logistic regression model that adopts all features (including the sensitive feature) in the training and prediction.
- **Adversarial de-biasing model** (abbreviated as Adv_Deb in the comparison)[170]: an in-processing model that proposes to maximize the predictive performance while minimizing the adversary's ability to predict the sensitive features.
- **Calibrated equal odds post-processing** (abbreviated as CEOP in the comparison)[110]: a post-processing model that proposes to minimize the error disparity among different groups indicated by the sensitive feature.
- **Disparate impact remover** (abbreviated as DIR in the comparison) [41]: a model that proposes to minimize the disparity in the outcome from different groups via pre-processing.
- **Reweighing method** [63]: a pre-processing method that eliminates the discrimination bias among different groups by reweighing and resampling the data.

We use three benchmark datasets to compare the model performance:

- **German credit data** from the UCI repository [34]: The data contains 1000 samples described by 20 features and the goal is to predict the credit risks (good or bad). The feature *personal status and sex* is used as the sensitive feature.
- **Compas**: The data includes 6167 samples described by 401 features with the outcome showing if each person was accused of a crime within two years. The feature *gender* is used as the sensitive feature in this data.
- **Bank marketing data**[135] from the UCI repository: The data consists of 45211 samples with 17 features. The goal is to predict whether a client will subscribe to a term deposit. The sensitive feature in this data is *age*.

We use the **classification accuracy** (percentage of correctly classified data in the testing set) and **balanced classification accuracy** (average of true positive rate and true negative rate) to evaluate the model prediction performance in the classification problem. Moreover, we adopt three

different metrics to evaluate the fairness among groups of population *w.r.t.* the sensitive feature in the data:

- **Absolute equal opportunity difference**: the absolute difference in true positive rate among different groups of population.
- **Absolute average odds difference**: the absolute difference in balanced classification accuracy among different groups of population.
- **Theil index**: proposed in [135] to measure the group or individual fairness. Here we report the absolute value of the Theil index, which is always positive. A close-to-zero Theil index indicates more fairness.

Features in the data are normalized to the range of $[0,\ 1]$. For each dataset, we randomly split the data into three sets: 60% for training set, 20% for validation set, and 20% for testing set, where the training set is used to train the model, validation set is used to tune the hyper-parameter, and the test is used to test the model performance. We run all comparing methods 5 times with 5 different random splits of the data and report the average performance with the standard deviation on the test set. For the methods involving a hyper-parameter, *i.e.,* the thresholding value in CEOP, DIR, and Reweighing method, we tune the hyper-parameter in the range of $\{0,\ 0.1,\ 0.2,\ \ldots,\ 1\}$ and use the best hyper-parameter achieving the best balanced classification accuracy on the validation set.

We implement the comparing methods via the AI Fairness 360 toolbox [11]. For our FAIAS model, we construct the predictor as a 4-layer neural network with 200 nodes in each layer. We adopt scaled exponential linear units (SELU)[72] as the activation function of the first 3 layers and the softmax function for the last layer. We use Adam optimizer [71] and set the learning rate as 0.0001. For the selector, we set it as a data-independent vector $\mathbf{w} = \frac{1}{1+e^{-\theta}} \in \mathbb{R}^d$ since we expect the selected features to be consistent among different samples. We use Tensorflow and Keras toolbox for implementing our code and run the algorithm on a machine with one Titan X Pascal GPU.

We first compare the model performance on the classification problems and summarize the results in Figure 20. The results show that our FAIAS model achieves the best classification result *w.r.t.* both the accuracy and the balanced accuracy, which indicate that the optimization on both the data and model perspective is successful in guaranteeing the prediction performance such that

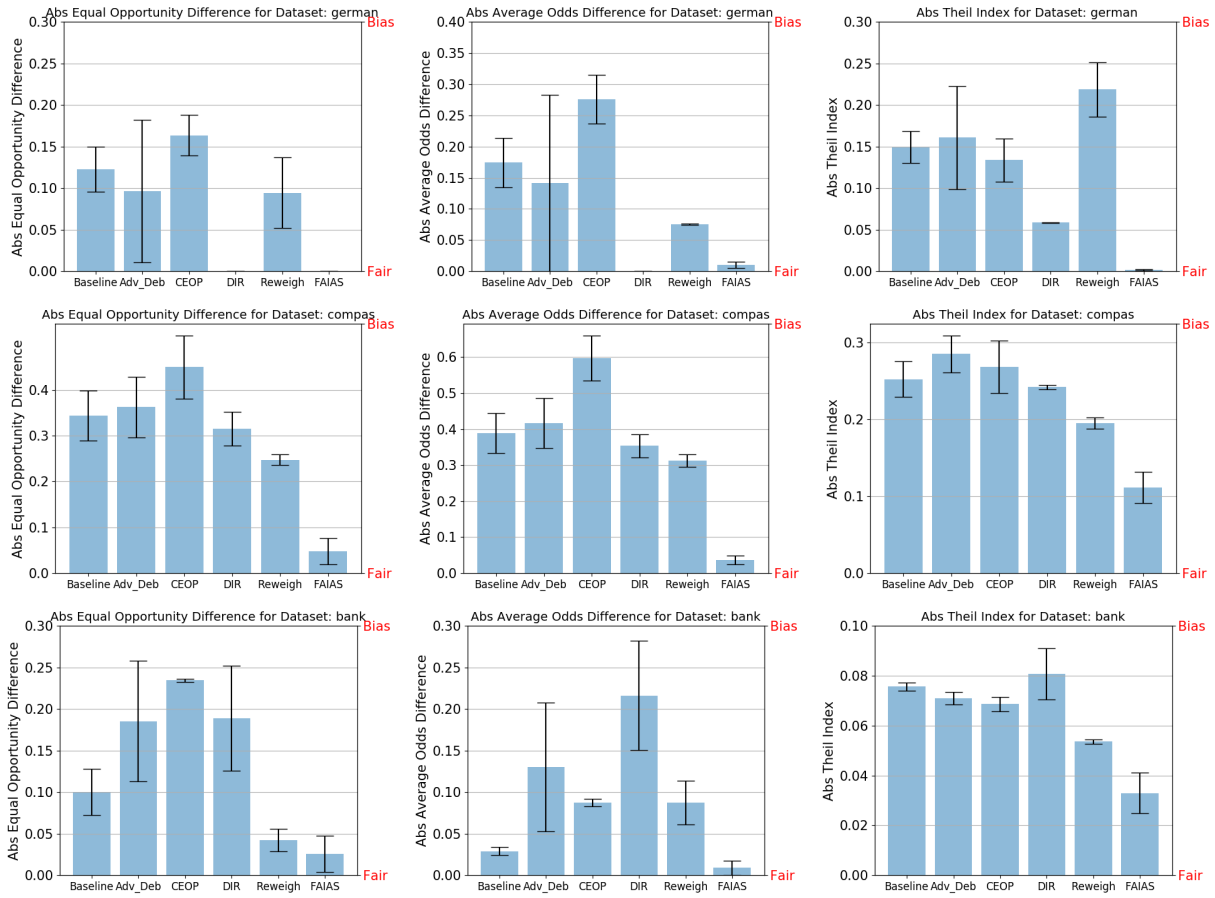

Figure 21: Comparison of prediction fairness via absolute equal opportunity difference, absolute average odds difference, and Theil index on three benchmark datasets.

imposing the fairness constraints does not sacrifice the classification performance. We also use the three fairness metrics to evaluate if FAIAS improves the prediction fairness by rendering equal prediction performance among different group of population. From the results in Figure 21, we can notice that FAIAS achieves equivalent or better results *w.r.t.* all three measurement metrics on the three benchmark datasets, such that the feature sampling via an adversarial network is able to eliminate the sensitive information and forces the prediction performance to be equalized among different groups of population.

Algorithm 4 Optimization Algorithm of FAIAS Model

Input data set $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, learning rate α_θ and α_ϕ.

Output

Initialize parameter θ and ϕ randomly.

while not converge **do**

for t = $1,\ 2,\ \ldots,\ n_b$ **do**

for $(\mathbf{x}_{t_i}, \mathbf{y}_{t_i})$ in the t-th mini-batch $\mathcal{Z}_t$ **do**

1. Calculate the selection probability vector

$$g^\theta(\mathbf{x}_{t_i}) = [p_{t_i}^1,\ p_{t_i}^2,\ \ldots,\ p_{t_i}^d].$$

2. Sample the selection vector $\mathbf{s}_{t_i} \in \mathbb{R}^d$ with

$$\mathbf{s}_{t_i}^j \sim Bernoulli(p_{t_i}^j), \quad \text{for } j = 1,\ ,2,\ \ldots,\ d.$$

3. Calculate

$$\hat{l}_{sen}(\mathbf{x}_{t_i}, \mathbf{s}_{t_i}) = f^\phi(\mathbf{x}_{t_i}, \mathbf{s}_{t_i} \cup \{k\}) - f^\phi(\mathbf{x}_{t_i}, \mathbf{s}_{t_i}).$$

end for

4. Update the parameter θ with gradient ascent

$$\theta \leftarrow \theta + \frac{\alpha_\theta}{n_b} \sum_i \hat{l}_{sen}(\mathbf{x}_{t_i}, \mathbf{s}_{t_i}) \nabla_\theta \log \pi_\theta(\mathbf{x}_{t_i}, \mathbf{s}_{t_i}).$$

5. Update the parameter ϕ with gradient descent

$$\phi \leftarrow \phi - \frac{\alpha_\phi}{n_b} \sum_i \frac{\hat{l}_{sen}(\mathbf{x}_{t_i}, \mathbf{s}_{t_i}) \nabla_\phi \hat{l}_{sen}(\mathbf{x}_{t_i}, \mathbf{s}_{t_i})}{||\hat{l}_{sen}(\mathbf{x}_{t_i}, \mathbf{s}_{t_i})||} - \frac{\alpha_\phi}{n_b} \sum_i \sum_{l=1}^{c} y_l \frac{\nabla_\phi f_l^\phi(\mathbf{x}_{t_i}, \mathbf{s}_{t_i})}{f_l^\phi(\mathbf{x}_{t_i}, \mathbf{s}_{t_i})}.$$

end for

end while

7.0 Conclusion

In this thesis, we propose several new nonlinear machine learning models to address the challenges in different data scales. We first address the overfitting and less interpretability problem of nonlinear kernel methods with a novel additive model (FNAM) for QTL identification. The framework of FNAM can be easily adapted to depict the non-linear associations between SNPs and brain endophenotypes. The experimental results on the ADNI cohort indicated the promising performance of FNAM. In particular, we not only identified some SNPs validated in the previous literature, but also found new SNPs with potential risk for Alzheimer's. These empirical studies validate the effectiveness of our approach, and provide insights into the genetic causal relationships as well as early detection of neurological disorders. We also derived the generalization error bound of FNAM under a general assumption, *i.e.,* m-dependent observations, thus is suitable to many other biological applications.

Next, we proposed a novel group sparse additive machine (GroupSAM) by incorporating the group sparsity into the additive classification model in reproducing kernel Hilbert space. By developing the error analysis technique with data dependent hypothesis space, we obtain the generalization error bound of the proposed GroupSAM, which demonstrates our model can achieve satisfactory learning rate under mild conditions. Experimental results on both synthetic and real-world benchmark datasets validate the algorithmic effectiveness and support our learning theory analysis. In the future, it is interesting to investigate the learning performance of robust group sparse additive machines with loss functions induced by quantile regression [23, 88].

What's more, we propose several new deep learning structures for large-scale machine learning. We design a novel conditional generative model (GGAN) as well as a novel deep generative model (SemiGAN) for gene expression inference. Compared with previous deep learning models considering minimum squared error loss that render blurry results, our model employed the coupled adversarial loss and ℓ_1-norm loss to make the regression results sharp and realistic. We validated our model on the inference of two different datasets, GEO and GTEx, and found consistent and significant improvements over all the counterparts. Moreover, we looked into the role of landmark genes in the prediction and identified different relevance pattern, which provided in-

sights into the relations among gene regulatory networks. In the future, we will investigate how to incorporate the profiles with only landmark gene measurement available using semi-supervised framework. Also, it would be interesting to employ the cluster structure among profile samples in the prediction to strengthen the inference of target gene expression.

We also analyze the longitudinal data with a novel Temporal-GAN model for MCI conversion prediction. Our model considered the data at all time points along the disease progression and uncovered the temporal correlation structure among the neuroimaing data at adjacent time points. We also constructed a generative model to produce more reliable data to strengthen the training process. Our model illustrated superiority in the experiments on the ADNI data. Moreover, we propose a general framework for building additive interpretable deep neural network and construct for direct understanding of the important neuroimaging features in the prediction.

Last but not, we propose a new adversarial network FAIAS for improving prediction fairness. We formulate our model from both the data perspective and the model perspective. Our FAIAS model consists of two components: a selector function and a prediction function, where the selector function is optimized on the data perspective to select the features containing only non-sensitive information and the prediction function is optimized from the model perspective to minimize the marginal contribution of the sensitive feature and also improve the prediction performance. We conduct extensive experiments on three benchmark datasets and validate that our FAIAS model outperforms all related methods *w.r.t.* both the prediction performance as well and fairness metrics.

Bibliography

[1] Julius Adebayo, Justin Gilmer, Ian Goodfellow, and Been Kim. Local explanation methods for deep neural networks lack sensitivity to parameter values. In *ICLR Workshop*, 2018.

[2] Tameem Adel, Isabel Valera, Zoubin Ghahramani, and Adrian Weller. One-network adversarial fairness. In *AAAI*, 2019.

[3] Saleema Amershi, Max Chickering, Steven M Drucker, Bongshin Lee, Patrice Simard, and Jina Suh. Modeltracker: Redesigning performance analysis tools for machine learning. In *CHI*, pages 337–346. ACM, 2015.

[4] Marco Ancona, Enea Ceolini, Cengiz Oztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *ICLR*, 2018.

[5] Liana G Apostolova, Mona Beyer, Amity E Green, Kristy S Hwang, Jonathan H Morra, Yi-Yu Chou, Christina Avedissian, Dag Aarsland, Carmen C Janvin, Jan P Larsen, et al. Hippocampal, caudate, and ventricular changes in parkinson's disease with and without dementia. *Movement Disorders*, 25(6):687–695, 2010.

[6] J. Ashburner and K. J. Friston. Voxel-based morphometry–the methods. *Neuroimage*, 11:805–821, 2000.

[7] Dimitrios Avramopoulos. Genetics of alzheimer's disease: recent advances. *Genome Med.*, 1(3):34, 2009.

[8] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[9] P. L. Bartlett, M. I. Jordan, and J. D. Mcauliffe. Convexity, classification and risk bounds. *J. Amer. Statist. Assoc.*, 101(473):138–156, 2006.

[10] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504*, 2017.

[11] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

[12] Mostapha Benhenda. Chemgan challenge for drug discovery: can ai reproduce natural chemical diversity? *arXiv preprint arXiv:1708.08227*, 2017.

[13] L. Bertram, M. B. McQueen, et al. Systematic meta-analyses of Alzheimer disease genetic association studies: the AlzGene database. *Nat Genet*, 39(1):17–23, 2007.

[14] Alexandre Calon, Enza Lonardo, Antonio Berenguer-Llergo, Elisa Espinet, Xavier Hernando-Momblona, Mar Iglesias, Marta Sevillano, Sergio Palomo-Ponce, Daniele VF Tauriello, Daniel Byrom, et al. Stromal gene expression defines poor-prognosis subtypes in colorectal cancer. *Nature genetics*, 47(4):320–329, 2015.

[15] Elisa Canu, Donald G McLaren, Michele E Fitzgerald, Barbara B Bendlin, Giada Zoccatelli, Franco Alessandrini, Francesca B Pizzini, Giuseppe K Ricciardi, Alberto Beltramello, Sterling C Johnson, et al. Mapping the structural brain changes in alzheimer's disease: the independent contribution of two imaging modalities. *Journal of Alzheimer's Disease*, 26(s3):263–274, 2011.

[16] Dennis Chan, Nick C Fox, Rachael I Scahill, William R Crum, Jennifer L Whitwell, Guy Leschziner, Alex M Rossor, John M Stevens, Lisa Cipolotti, and Martin N Rossor. Patterns of temporal lobe atrophy in semantic dementia and alzheimer's disease. *Annals of neurology*, 49(4):433–442, 2001.

[17] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27):1–27, 2011.

[18] D. R. Chen, Q. Wu, Y. Ying, and D. X. Zhou. Support vector machine soft margin classifiers: error analysis. *J. Mach. Learn. Res.*, 5:1143–1175, 2004.

[19] H. Chen, Z. Pan, L. Li, and Y.Y. Tang. Learning rates of coefficient-based regularized classifier for density level detection. *Neural Comput.*, 25(4):1107–1121, 2013.

[20] Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, and Xiaohui Xie. Gene expression inference with deep learning. *Bioinformatics*, 32(12):1832–1839, 2016.

[21] LI Chongxuan, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 4091–4101, 2017.

[22] A. Christmann and R. Hable. Consistency of support vector machines using additive kernels for additive models. *Comput. Stat. Data Anal.*, 56:854–873, 2012.

[23] A. Christmann and D. X. Zhou. Learning rates for the risk of kernel based quantile regression estimators in additive models. *Anal. Appl.*, 14(3):449–477, 2016.

[24] Lingyang Chu, Xia Hu, Juhua Hu, Lanjun Wang, and Jian Pei. Exact and consistent interpretation for piecewise linear neural networks: A closed form solution. *arXiv preprint arXiv:1802.06259*, 2018.

[25] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning (ICML)*, pages 160–167. ACM, 2008.

[26] A Convit, J De Asis, MJ De Leon, CY Tarshish, S De Santi, and H Rusinek. Atrophy of the medial occipitotemporal, inferior, and middle temporal gyri in non-demented elderly predict decline to alzheimers disease. *Neurobiology of aging*, 21(1):19–26, 2000.

[27] Johnathan Cooper-Knock, Pamela J Shaw, and Janine Kirby. The widening spectrum of c9orf72-related disease; genotype/phenotype correlations and potential modifiers of clinical phenotype. *Acta. Neuropathol.*, 127(3):333–345, 2014.

[28] F. Cucker and S. Smale. On the mathematical foundations of learning. *Bull. Amer. Soc.*, 39(1):1–49, 2001.

[29] Felipe Cucker and Ding Xuan Zhou. *Learning theory: an approximation theory viewpoint*, volume 24. Cambridge University Press, 2007.

[30] Spyros Darmanis, Steven A Sloan, Ye Zhang, Martin Enge, Christine Caneda, Lawrence M Shuer, Melanie G Hayden Gephart, Ben A Barres, and Stephen R Quake. A survey of human brain transcriptome diversity at the single cell level. *Proceedings of the National Academy of Sciences*, 112(23):7285–7290, 2015.

[31] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems (NIPS)*, pages 1486–1494, 2015.

[32] Pietro Di Lena, Ken Nagata, and Pierre Baldi. Deep architectures for protein contact map prediction. *Bioinformatics*, 28(19):2449–2457, 2012.

[33] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[34] Dheeru Dua and Casey Graff. UCI machine learning repository, 2019.

[35] Qiaonan Duan, Corey Flynn, Mario Niepel, Marc Hafner, Jeremy L Muhlich, Nicolas F Fernandez, Andrew D Rouillard, Christopher M Tan, Edward Y Chen, Todd R Golub, et al. Lincs canvas browser: interactive web app to query, browse and interrogate lincs l1000 gene expression signatures. *Nucleic acids research*, 42(W1):W449–W460, 2014.

[36] Ron Edgar, Michael Domrachev, and Alex E Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210, 2002.

[37] D. Edmunds and H. Triebel. *Function Spaces, Entropy Numbers, Differential Operators*. Cambridge Univ. Press, Cambridge, U.K., 1996.

[38] Dumitru Erhan, Y Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *Technical Report, Univerist de Montral*, 1341(3):1–13, 2009.

[39] Rasool Fakoor, Faisal Ladhak, Azade Nazi, and Manfred Huber. Using deep learning to enhance cancer diagnosis and classification. In *Proceedings of the International Conference on Machine Learning*, 2013.

[40] Maryam Fazel. *Matrix rank minimization with applications*. PhD thesis, Electrical Engineering Department, Stanford University, 2002.

[41] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015.

[42] Greg Finak, Nicholas Bertos, Francois Pepin, Svetlana Sadekova, Margarita Souleimanova, Hong Zhao, Haiying Chen, Gulbeyaz Omeroglu, Sarkis Meterissian, Atilla Omeroglu, et al. Stromal gene expression predicts clinical outcome in breast cancer. 14(5):518.

[43] Samuele Fiorini, Alessandro Verri, Annalisa Barla, Andrea Tacchino, and Giampaolo Brichetto. Temporal prediction of multiple sclerosis evolution from patient-centered outcomes. In *Machine Learning for Healthcare Conference*, pages 112–125, 2017.

[44] B. Fischl, D. H. Salat, et al. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33(3):341–55, 2002.

[45] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[46] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[47] Deviprasad R Gollapalli and Robert R Rando. All-trans-retinyl esters are the substrates for isomerization in the vertebrate visual cycle. *Biochemistry*, 42(19):5809–5818, 2003.

[48] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems (NIPS)*, pages 2672–2680, 2014.

[49] Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a" right to explanation". *arXiv preprint arXiv:1606.08813*, 2016.

[50] Thomas F Gross. Face recognition and own-ethnicity bias in black, east/southeast asian, hispanic, and white children. *Asian American Journal of Psychology*, 5(3):181, 2014.

[51] Xiaobo Guo, Ye Zhang, Wenhao Hu, Haizhu Tan, and Xueqin Wang. Inferring nonlinear gene regulatory networks from gene expression data based on distance correlation. *PloS one*, 9(2):e87446, 2014.

[52] Z. Guo and D. X. Zhou. Concentration estimates for learning with unbounded sampling. *Adv. Comput. Math.*, 38(1):207–223, 2013.

[53] Graham Heimberg, Rajat Bhatnagar, Hana El-Samad, and Matt Thomson. Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. *Cell systems*, 2(4):239–250, 2016.

[54] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[55] Kun Hu, Yijue Wang, Kewei Chen, Likun Hou, and Xiaoqun Zhang. Multi-scale features extraction from baseline structure mri for mci patient classification and ad early diagnosis. *Neurocomputing*, 175:132–145, 2016.

[56] Chaorui Huang, Lars-Olof Wahlund, Leif Svensson, Bengt Winblad, and Per Julin. Cingulate cortex hypoperfusion predicts alzheimer's disease in mild cognitive impairment. *BMC neurology*, 2(1):9, 2002.

[57] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017.

[58] J. Huang, J. Horowitz, and F. Wei. Variable selection in nonparametric additive models. *Ann. Statist.*, 38(4):2282–2313, 2010.

[59] Zhouyuan Huo, Dinggang Shen, and Heng Huang. Genotype-phenotype association study via new multi-task learning model. *Twenty-Third Pacific Symposium on Biocomputing (PSB 2018)*, pages 353–364, 2018.

[60] Boris Igelnik and Yoh-Han Pao. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE/ACM Trans. Netw.*, 6(6):1320–1329, 1995.

[61] S. Mattu J. Angwin, J. Larson and L. Kirchner. Machine bias: Theres software used across the country to predict future criminals. and its biased against blacks. *ProPublica*, 2016.

[62] Noor Jehan Kabani. 3d anatomical atlas of the human brain. *Neuroimage*, 7:P–0717, 1998.

[63] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

[64] K. Kandasamy and Y. Yu. Additive approximation in high dimensional nonparametric regression via the salsa. In *ICML*, 2016.

[65] Cyriac Kandoth, Michael D McLellan, Fabio Vandin, Kai Ye, Beifang Niu, Charles Lu, Mingchao Xie, Qunyuan Zhang, Joshua F McMichael, Matthew A Wyczalkowski,

et al. Mutational landscape and significance across 12 major cancer types. *Nature*, 502(7471):333, 2013.

[66] Kejal Kantarci, Clifford R Jack Jr, Yue Cheng Xu, Norberg G Campeau, Peter C OBrien, Glenn E Smith, Robert J Ivnik, Bradley F Boeve, Emre Kokmen, Eric G Tangalos, et al. Mild cognitive impairment and alzheimer disease: regional diffusivity of water. *Radiology*, 219(1):101–107, 2001.

[67] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[68] Alexandra B Keenan, Sherry L Jenkins, Kathleen M Jagodnik, Simon Koplev, Edward He, Denis Torre, Zichen Wang, Anders B Dohlman, Moshe C Silverstein, Alexander Lachmann, et al. The library of integrated network-based cellular signatures nih program: system-level cataloging of human cells response to perturbations. *Cell systems*, 2017.

[69] Jaedeok Kim and Jingoo Seo. Human understandable explanation extraction for black-box classification models based on matrix factorization. *arXiv preprint arXiv:1709.06201*, 2017.

[70] Pieter-Jan Kindermans, Kristof T Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. Learning how to explain neural networks: patternnet and patternattribution. *arXiv preprint arXiv:1705.05598*, 2017.

[71] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[72] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In *Advances in neural information processing systems*, pages 971–980, 2017.

[73] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017.

[74] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[75] Helen F Ladd. Evidence on discrimination in mortgage lending. *Journal of Economic Perspectives*, 12(2):41–62, 1998.

[76] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *KDD*, pages 1675–1684. ACM, 2016.

[77] Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat. Rev. Genet.*, 11(10):733–739, 2010.

[78] Luís Lemos, Dina Silva, Manuela Guerreiro, Isabel Santana, Alexandre de Mendonça, Pedro Tomás, and Sara C Madeira. Discriminating alzheimer's disease from mild cognitive impairment using neuropsychological data. *Age (M±SD)*, 70(8.4):73–3, 2012.

[79] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, David Madigan, et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *Ann. Appl. Stat.*, 9(3):1350–1371, 2015.

[80] Y. Li, C. J. Willer, et al. MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genet Epidemiol*, 34(8):816–34, 2010.

[81] Winnie S Liang, Eric M Reiman, Jon Valla, Travis Dunckley, Thomas G Beach, Andrew Grover, Tracey L Niedzielko, Lonnie E Schneider, Diego Mastroeni, Richard Caselli, et al. Alzheimer's disease is associated with reduced expression of energy metabolism genes in posterior cingulate neurons. *Proc. Natl. Acad. Sci. U.S.A.*, 105(11):4441–4446, 2008.

[82] M. Lichman. UCI machine learning repository, 2013.

[83] Y. Lin and H. H. Zhang. Component selection and smoothing in smoothing spline analysis of variance models. *Ann. Statist.*, 34(5):2272–2297, 2006.

[84] Lydia Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *International Conference on Machine Learning*, pages 3156–3164, 2018.

[85] Siqi Liu, Sidong Liu, Weidong Cai, Hangyu Che, Sonia Pujol, Ron Kikinis, Dagan Feng, Michael J Fulham, et al. Multimodal neuroimaging feature learning for multiclass diagnosis of alzheimer's disease. *IEEE Transactions on Biomedical Engineering*, 62(4):1132–1140, 2015.

[86] Wei Liu, Jun Wang, and Shih-Fu Chang. Robust and scalable graph-based semisupervised learning. *Proceedings of the IEEE*, 100(9):2624–2638, 2012.

[87] John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, et al. The genotype-tissue expression (gtex) project. *Nature genetics*, 45(6):580, 2013.

[88] S. Lv, H. Lin, H. Lian, and J. Huang. Oracle inequalities for sparse additive quantile regression in reproducing kernel hilbert space. *Ann. Statist.*, preprint, 2017.

[89] James Lyons, Abdollah Dehzangi, Rhys Heffernan, Alok Sharma, Kuldip Paliwal, Abdul Sattar, Yaoqi Zhou, and Yuedong Yang. Predicting backbone cα angles and dihedrals from protein sequences by stacked sparse auto-encoder deep neural network. *Journal of computational chemistry*, 35(28):2040–2046, 2014.

[90] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning (ICML)*, volume 30, 2013.

[91] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821. IEEE, 2017.

[92] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[93] Marnix H Medema, Renzo Kottmann, Pelin Yilmaz, Matthew Cummings, John B Biggins, Kai Blin, Irene De Bruijn, Yit Heng Chooi, Jan Claesen, R Cameron Coates, et al. Minimum information about a biosynthetic gene cluster. *Nature chemical biology*, 11(9):625, 2015.

[94] L. Meier, S. van de Geer, and P. Buehlmann. High-dimensional additive modeling. *Ann. Statist.*, 37(6B):3779–3821, 2009.

[95] A Meyer-Lindenberg, T Nichols, JH Callicott, J Ding, B Kolachana, J Buckholtz, VS Mattay, M Egan, and DR Weinberger. Impact of complex genetic variation in comt on human brain function. *Molecular psychiatry*, 11(9):867, 2006.

[96] Dharmendra S Modha and Elias Masry. Minimum complexity regression estimation with weakly dependent observations. *IEEE Trans. Inf. Theory*, 42(6):2133–2145, 1996.

[97] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognit.*, 65:211–222, 2017.

[98] D Mouttet, M Laé, M Caly, D Gentien, S Carpentier, H Peyro-Saint-Paul, A Vincent-Salomon, R Rouzier, B Sigal-Zafrani, X Sastre-Garau, et al. Estrogen-receptor, progesterone-receptor and her2 status determination in invasive breast cancer. concordance between immuno-histochemistry and mapquant microarray based assay. *PloS one*, 11(2):e0146474, 2016.

[99] Hussein Mouzannar, Mesrob I Ohannessian, and Nathan Srebro. From fair decision making to social equality. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 359–368. ACM, 2019.

[100] Bradlee D Nelms, Levi Waldron, Luis A Barrera, Andrew W Weflen, Jeremy A Goettel, Guoji Guo, Robert K Montgomery, Marian R Neutra, David T Breault, Scott B Snapper, et al. Cellmapper: rapid and accurate inference of gene expression in difficult-to-isolate cell types. *Genome biology*, 17(1):201, 2016.

[101] Milap A Nowrangi and Paul B Rosenberg. The fornix in mild cognitive impairment and alzheimers disease. *Frontiers in aging neuroscience*, 7:1, 2015.

[102] Vasilis Ntranos, Govinda M Kamath, Jesse M Zhang, Lior Pachter, and N Tse David. Fast and accurate single-cell rna-seq analysis by clustering of transcript-compatibility counts. *Genome biology*, 17(1):112, 2016.

[103] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization.

[104] Kenjiro Ono, Yuji Yoshiike, Akihiko Takashima, Kazuhiro Hasegawa, Hironobu Naiki, and Masahito Yamada. Vitamin a exhibits potent antiamyloidogenic and fibril-destabilizing effects in vitro. *Exp. Neurol.*, 189(2):380–392, 2004.

[105] Andreas Papassotiropoulos, M Axel Wollmer, Magdalini Tsolaki, Fabienne Brunner, Dimitra Molyva, Dieter Lütjohann, Roger M Nitsch, and Christoph Hock. A cluster of cholesterol-related genes confers susceptibility for alzheimer's disease. *J. Clin. Psychiatry*, 66(7):940–947, 2005.

[106] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

[107] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544.

[108] Jiska S Peper, Rachel M Brouwer, Dorret I Boomsma, René S Kahn, Hulshoff Pol, and E Hilleke. Genetic influences on human brain structure: a review of brain imaging studies in twins. *Human brain mapping*, 28(6):464–473, 2007.

[109] Isabelle Perrault, Sylvain Hanein, Sylvie Gerber, Fabienne Barbet, Dominique Ducroq, Helene Dollfus, Christian Hamel, Jean-Louis Dufier, Arnold Munnich, Josseline Kaplan, et al. Retinal dehydrogenase 12 (rdh12) mutations in leber congenital amaurosis. *Am. J. Hum. Genet.*, 75(4):639–646, 2004.

[110] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In *NIPS*, pages 5684–5693, 2017.

[111] Randall J Pruim, Ryan P Welch, Serena Sanna, Tanya M Teslovich, Peter S Chines, Terry P Gliedt, Michael Boehnke, Gonçalo R Abecasis, and Cristen J Willer. Locuszoom: regional visualization of genome-wide association scan results. *Bioinformatics*, 26(18):2336–2337, 2010.

[112] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *NIPS*, pages 1313–1320, 2009.

[113] G. Raskutti, M. Wainwright, and B. Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *J. Mach. Learn. Res.*, 13:389–427, 2012.

[114] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman. Sparse additive models. *J. Royal. Statist. Soc B.*, 71:1009–1030, 2009.

[115] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *KDD*, pages 1135–1144. ACM, 2016.

[116] Ekaterina Rogaeva, Yan Meng, Joseph H Lee, Yongjun Gu, Toshitaka Kawarai, Fanggeng Zou, Taiichi Katayama, Clinton T Baldwin, Rong Cheng, Hiroshi Hasegawa, et al. The neuronal sortilin-related receptor sorl1 is genetically associated with alzheimer disease. *Nature Genet.*, 39(2):168–177, 2007.

[117] Adriana Romero, Pierre Luc Carrier, Akram Erraqabi, Tristan Sylvain, Alex Auvolat, Etienne Dejoie, Marc-André Legault, Marie-Pierre Dubé, Julie G Hussin, and Yoshua Bengio. Diet networks: Thin parameters for fat genomic. *arXiv preprint arXiv:1611.09340*, 2016.

[118] Natalie S Ryan, Jennifer M Nicholas, Philip SJ Weston, Yuying Liang, Tammaryn Lashley, Rita Guerreiro, Gary Adamson, Janna Kenny, Jon Beck, Lucia Chavez-Gutierrez, et al. Clinical phenotype and genetic associations in autosomal dominant familial alzheimers disease: a case series. *Lancet Neurol.*, 15(13):1326–1335, 2016.

[119] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2234–2242, 2016.

[120] Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 901–909, 2016.

[121] A. J. Saykin, L. Shen, et al. Alzheimer's disease neuroimaging initiative biomarkers as quantitative phenotypes: Genetics core aims, progress, and plans. *Alzheimers Dement*, 6(3):265–73, 2010.

[122] Wouter F Schmidt, Martin A Kraaijveld, and Robert PW Duin. Feedforward neural networks with random weights. In *Proc. 11th IAPR Int. Conf. Pattern Recognition Methodology Systems*, pages 1–4. IEEE, 1992.

[123] Daniel Schmitter, Alexis Roche, Bénédicte Maréchal, Delphine Ribes, Ahmed Abdulkadir, Meritxell Bach-Cuadra, Alessandro Daducci, Cristina Granziera, Stefan Klöppel, Philippe Maeder, et al. An evaluation of volume-based morphometry for prediction of mild cognitive impairment and alzheimer's disease. *NeuroImage: Clinical*, 7:7–17, 2015.

[124] Sheel Shah, Eric Lubeck, Wen Zhou, and Long Cai. In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. *Neuron*, 92(2):342–357, 2016.

[125] Dinggang Shen and Christos Davatzikos. Hammer: hierarchical attribute matching mechanism for elastic registration. *Medical Imaging, IEEE Transactions on*, 21(11):1421–1439, 2002.

[126] L. Shen, S. Kim, et al. Whole genome association study of brain-wide imaging phenotypes for identifying quantitative trait loci in MCI and AD: A study of the ADNI cohort. *Neuroimage*, 53(3):1051–63, 2010.

[127] Jiajun Shi, Sizhong Zhang, Mouni Tang, Xiehe Liu, Tao Li, Haiying Han, Yingcheng Wang, Yangbo Guo, Jinghua Zhao, Hai Li, et al. Possible association between cys311ser polymorphism of paraoxonase 2 gene and late-onset alzheimer's disease in chinese. *Brain Res. Mol. Brain Res.*, 120(2):201–204, 2004.

[128] L. Shi. Learning theory estimates for coefficient-based regularized regression. *Appl. Comput. Harmon. Anal.*, 34(2):252–265, 2013.

[129] L. Shi, Y. Feng, and D. X. Zhou. Concentration estimates for learning with ℓ_1-regularizer and data dependent hypothesis spaces. *Appl. Comput. Harmon. Anal.*, 31(2):286–302, 2011.

[130] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*, 2017.

[131] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[132] John G Sled, Alex P Zijdenbos, and Alan C Evans. A nonparametric method for automatic correction of intensity nonuniformity in mri data. *Medical Imaging, IEEE Transactions on*, 17(1):87–97, 1998.

[133] Gary W Small, Linda M Ercoli, Daniel HS Silverman, S-C Huang, Scott Komo, Susan Y Bookheimer, Helen Lavretsky, Karen Miller, Prabha Siddarth, Natalie L Rasgon, et al. Cerebral metabolic and cognitive decline in persons at genetic risk for alzheimer's disease. *Proc. Natl. Acad. Sci. U.S.A.*, 97(11):6037–6042, 2000.

[134] Nora K Speicher and Nico Pfeifer. Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery. *Bioinformatics*, 31(12):i268–i275, 2015.

[135] Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. A unified approach to quantifying algorithmic unfairness: Measuring individual &group unfairness via inequality indices. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2239–2248. ACM, 2018.

[136] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[137] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[138] I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

[139] Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B.*, 58(1):267–288, 1996.

[140] A. B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *Ann. Statis.*, 32:135–166, 2004.

[141] Wiesje M van der Flier, Yolande AL Pijnenburg, Nick C Fox, and Philip Scheltens. Early-onset versus late-onset alzheimer's disease: the case of the missing apoe ϵ4 allele. *Lancet Neurol.*, 10(3):280–288, 2011.

[142] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

[143] Mathukumalli Vidyasagar. *Learning and generalisation: with applications to neural networks*. New York: Springer-Verlag, 2013.

[144] Maria Vounou, Thomas E Nichols, Giovanni Montana, Alzheimer's Disease Neuroimaging Initiative, et al. Discovering genetic associations with high-dimensional neuroimaging phenotypes: a sparse reduced-rank regression approach. *Neuroimage*, 53(3):1147–1159, 2010.

[145] Fulton Wang and Cynthia Rudin. Falling rule lists. In *AISTATS*, pages 1013–1022, 2015.

[146] Hua Wang, Feiping Nie, Heng Huang, Sungeun Kim, Kwangsik Nho, Shannon L Risacher, Andrew J Saykin, Li Shen, and Alzheimer's Disease Neuroimaging Initiative. Identifying quantitative trait loci via group-sparse multitask regression and feature selection: an imaging genetics study of the adni cohort. *Bioinformatics*, 28(2):229–237, 2011.

[147] Hua Wang, Feiping Nie, Heng Huang, Sungeun Kim, Kwangsik Nho, Shannon L Risacher, Andrew J Saykin, Li Shen, Alzheimer's Disease Neuroimaging Initiative, et al. Identifying quantitative trait loci via group-sparse multitask regression and feature selection: an imaging genetics study of the adni cohort. *Bioinformatics*, 28(2):229–237, 2012.

[148] Hua Wang, Feiping Nie, Heng Huang, Jingwen Yan, Sungeun Kim, Kwangsik Nho, Shannon L Risacher, Andrew J Saykin, Li Shen, Alzheimer's Disease Neuroimaging Initiative, et al. From phenotype to genotype: an association study of longitudinal phenotypic markers to alzheimer's disease relevant snps. *Bioinformatics*, 28(18):i619–i625, 2012.

[149] Xiaoqian Wang, Hong Chen, Weidong Cai, Dinggang Shen, and Heng Huang. Regularized modal regression with applications in cognitive impairment prediction. In *NIPS*, pages 1447–1457, 2017.

[150] Xiaoqian Wang, Kefei Liu, Jingwen Yan, Shannon L Risacher, Andrew J Saykin, Li Shen, Heng Huang, et al. Predicting interrelated alzheimers disease outcomes via new self-learned structured low-rank model. In *International Conference on Information Processing in Medical Imaging*, pages 198–209. Springer, 2017.

[151] Xiaoqian Wang, Jingwen Yan, Xiaohui Yao, Sungeun Kim, Kwangsik Nho, Shannon L. Risacher, Andrew J. Saykin, Li Shen, and Heng Huang. Longitudinal genotype-phenotype

association study via temporal structure auto-learning predictive model. *The 21st Annual International Conference on Research in Computational Molecular Biology (RECOMB 2017)*, pages 287–302, 2017.

[152] Yaping Wang, Jingxin Nie, Pew-Thian Yap, Gang Li, Feng Shi, Xiujuan Geng, Lei Guo, Dinggang Shen, Alzheimer's Disease Neuroimaging Initiative, et al. Knowledge-guided robust mri brain extraction for diverse large-scale neuroimaging studies on humans and non-human primates. *PloS one*, 9(1):e77810, 2014.

[153] Yaping Wang, Jingxin Nie, Pew-Thian Yap, Feng Shi, Lei Guo, and Dinggang Shen. Robust deformable-surface-based skull-stripping for large-scale studies. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2011*, pages 635–642. Springer, 2011.

[154] Stephen C Waring and Roger N Rosenberg. Genome-wide association studies in alzheimer disease. *Arch. Neurol.*, 65(3):329–334, 2008.

[155] Larry Wasserman and John D Lafferty. Statistical analysis of semi-supervised regression. In *Advances in Neural Information Processing Systems*, pages 801–808.

[156] Rizhen Wei, Chuhan Li, Noa Fogelson, and Ling Li. Prediction of conversion from mild cognitive impairment to alzheimer's disease using mri and structural network features. *Frontiers in aging neuroscience*, 8, 2016.

[157] Michael W Weiner, Dallas P Veitch, Paul S Aisen, Laurel A Beckett, Nigel J Cairns, Robert C Green, Danielle Harvey, Clifford R Jack, William Jagust, Enchi Liu, et al. The alzheimer's disease neuroimaging initiative: a review of papers published since its inception. *Alzheimers Dement.*, 9(5):e111–e194, 2013.

[158] Q. Wu, Y. Ying, and D. X. Zhou. Multi-kernel regularized classfiers. *J. Complexity*, 23:108–134, 2007.

[159] Q. Wu and D. X. Zhou. Svm soft margin classifiers: linear programming versus quadratic programming. *Neural Comput.*, 17:1160–1187, 2005.

[160] Wenjun Yan, Jianchao Wei, Xufang Deng, Zixue Shi, Zixiang Zhu, Donghua Shao, Beibei Li, Shaohui Wang, Guangzhi Tong, and Zhiyong Ma. Transcriptional analysis of immune-related gene expression in p53-deficient mice with increased susceptibility to influenza a virus infection. *BMC medical genomics*, 8(1):52, 2015.

[161] L. Yang, S. Lv, and J. Wang. Model-free variable selection in reproducing kernel hilbert space. *J. Mach. Learn. Res.*, 17:1–24, 2016.

[162] Tao Yang, Jie Wang, Qian Sun, Derrek P Hibar, Neda Jahanshad, Li Liu, Yalin Wang, Liang Zhan, Paul M Thompson, and Jieping Ye. Detecting genetic risk factors for alzheimer's disease in whole genome sequence data via lasso screening. In *Proc. IEEE Int. Symp. Biomed. Imaging*, pages 985–989. IEEE, 2015.

[163] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5485–5493, 2017.

[164] Muhammed A Yıldırım, Kwang-Il Goh, Michael E Cusick, Albert-László Barabási, and Marc Vidal. Drug-target network. *Nature biotechnology*, 25(10):1119–1126, 2007.

[165] J. Yin, X. Chen, and E.P. Xing. Group sparse additive models. In *ICML*, 2012.

[166] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variabels. *J. Royal. Statist. Soc B.*, 68(1):49–67, 2006.

[167] M. Yuan and D. X. Zhou. Minimax optimal rates of estimation in high dimensional additive models. *Ann. Statist.*, 44(6):2564–2593, 2016.

[168] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.

[169] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

[170] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340. ACM, 2018.

[171] T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Ann. Statist.*, 32:56–85, 2004.

[172] Yongyue Zhang, Michael Brady, and Stephen Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *Medical Imaging, IEEE Transactions on*, 20(1):45–57, 2001.

[173] T. Zhao and H. Liu. Sparse additive machine. In *AISTATS*, 2012.

[174] Leon Wenliang Zhong and James T Kwok. Efficient sparse modeling with automatic feature grouping. In *ICML*, 2011.

[175] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

[176] Xiaofeng Zhu, Heung-Il Suk, Dinggang Shen, and Heng Huang. Structured spare low-rank regression model for brain-wide and genome-wide associations. *18th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2016)*, pages 344–352, 2016.

[177] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

[178] Bin Zou, Luoqing Li, and Zongben Xu. The generalization performance of erm algorithm with strongly mixing observations. *Mach. Learn.*, 75(3):275–295, 2009.